

ANDRES JESSÉ PORFIRIO

**RECONHECIMENTO DAS CONFIGURAÇÕES DE MÃO DA
LIBRAS A PARTIR DE MALHAS 3D**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientador: Prof. Dr. Daniel Weingaertner

CURITIBA

2013

SUMÁRIO

RESUMO	iii
ABSTRACT	iv
1 INTRODUÇÃO	1
1.1 Definição do Problema	2
1.2 Objetivos	3
1.3 Justificativa	4
1.4 Desafios	5
1.5 Contribuições	5
1.6 Organização do Trabalho	6
2 FUNDAMENTAÇÃO TEÓRICA	7
2.1 Geração de Malhas 3D	8
2.1.1 Geração da Malha por Silhueta	8
2.1.2 Pré-Processamento das Malhas	11
2.2 Harmônicos Esféricos (Spherical Harmonics)	13
2.3 Support Vector Machine	18
3 REVISÃO BIBLIOGRÁFICA	22
3.1 Reconhecimento de Línguas de Sinais	22
3.1.1 Abordagem baseada em visualização	23
3.1.2 Abordagem baseada em sensores	25
3.2 Resumo dos Trabalhos	29
3.3 Bases de Dados	31
4 CLASSIFICAÇÃO DE MALHAS 3D DAS CONFIGURAÇÕES DE MÃO DA LIBRAS	33

4.1	Aquisição da Base de Dados	33
4.2	Reconstrução por Silhueta	43
4.3	Computação do Descritor Spherical Harmonics	46
4.4	Classificação 3D	46
4.5	Classificação 2D	50
5	RESULTADOS	53
5.1	Aquisição da Base de Dados	53
5.2	Reconstrução por Silhueta	55
5.3	Computação do Descritor Spherical Harmonics	55
5.4	Classificação	56
5.4.1	Esféricos Harmônicos em 61 Classes	57
5.4.2	Características 2D	57
5.4.3	Impacto da Suavização das Malhas na Classificação	58
5.4.4	Redução de Atributos	59
5.4.5	Análise das matrizes de confusão	60
5.5	Comparação de Resultados e Considerações	63
6	CONCLUSÃO	66
	BIBLIOGRAFIA	68
A	EXEMPLOS DE IMAGENS 2D DAS VISÕES FRONTAL E LATERAL DA MÃO.	74
B	EXEMPLOS DE MALHAS GERADAS POR SILHUETA	80

RESUMO

O reconhecimento automático de sinais é um processo importante para uma boa utilização dos meios de comunicação digitais por deficientes auditivos e, além disso, favorece a comunicação entre surdos e ouvintes que não compreendem a língua de sinais. A abordagem de reconhecimento de sinais utilizada neste trabalho baseia-se nos parâmetros globais da LIBRAS - língua brasileira de sinais: configuração de mão, locação ou ponto de articulação, movimento, orientação da palma da mão e expressão facial. A união de parâmetros globais forma sinais assim como fonemas formam palavras na língua falada.

Este trabalho apresenta uma forma de reconhecer um dos parâmetros globais da LIBRAS, a configuração de mão, a partir de malhas tridimensionais. A língua brasileira de sinais conta com 61 configurações de mão[16], este trabalho fez uso de uma base de dados contendo 610 vídeos de 5 usuários distintos em duas tomadas, totalizando 10 capturas para cada configuração de mão.

De cada vídeo foram extraídos manualmente dois quadros retratando as visões frontal e lateral da mão que, após segmentados e pré-processados, foram utilizados como entrada para o processamento de reconstrução 3D. A geração da malha 3D a partir das visões frontal e lateral da mão foi feita com o uso da técnica de reconstrução por silhueta[7].

O reconhecimento das configurações de mão a partir das malhas 3D foi feito com o uso do classificador SVM - Support Vector Machine. As características utilizadas para distinguir as malhas foram obtidas com o método Spherical Harmonics[25], um descritor de malhas 3D invariante à rotação, translação e escala. Os resultados atingiram uma taxa de acerto média de 98.52% com Ranking 5 demonstrando a eficiência do método.

ABSTRACT

Automatic recognition of Sign Language signs is an important process that enhances the quality of use of digital media by hearing impaired people. Additionally, sign recognition enables a way of communication between deaf and hearing people who do not understand Sign Language. The approach of sign recognition used in this work is based on the global parameters of LIBRAS (Brazilian Sign Language): hand configuration, location or point of articulation, movement, palm orientation and facial expression. These parameters are combined to comprise signs, in a similar manner that phonemes are used to form words in spoken (oral) language.

This paper presents a way to recognize one of the LIBRAS global parameters, the hand configuration, from 3D meshes. The Brazilian Sign Language has 61 hand configurations [16]. This work made use of a database containing 610 videos of 5 different users signing each hand configuration twice at distinct times, totaling 10 captures for each hand configuration.

Two pictures depicting the front and the side views of the hand were manually extracted from each video. These pictures were segmented and pre-processed, after which they were used as input to the 3D reconstruction processing. The generation of the 3D meshes from the front and side images of the hand configuration was done using the Shape from Silhouette technique[7].

The recognition of the hand configurations from the 3D meshes was done with the use of SVM classifier - Support Vector Machine. The characteristics used to distinguish the mesh were obtained using the Spherical Harmonics [25] method: a 3D mesh descriptor that is rotation, translation and scale invariant. Results achieved a hit rate average of 98.52% with Rank 5, demonstrating the efficiency of the method.

CAPÍTULO 1

INTRODUÇÃO

Línguas de sinais são utilizadas por deficientes auditivos como um dos principais meios de comunicação, obtenção de informações, educação, entre outros [2]. Não existe uma língua de sinais universal. Diferentes países utilizam diferentes línguas de sinais.

A linguagem de comunicação oficial utilizada pelos surdos no Brasil é chamada LIBRAS [39]. A criação da LIBRAS foi baseada na estrutura da língua de sinais francesa, possuindo similaridades com a americana e diversas línguas européias. A LIBRAS, como outras línguas de sinais, é composta por diversos por níveis linguísticos: fonologia, morfologia, sintaxe e semântica [30]. Portanto, não sendo a soletração do português [2].

Dada a grande quantidade de pessoas que utilizam a LIBRAS em seu dia-a-dia, torna-se necessária a existência de ferramentas que possibilitem sua utilização também em meios de comunicação digitais. A utilização da língua de sinais em sistemas digitais pode ser feita de várias maneiras, ou seja, inclusão de avatares animados para sintetização de sinais em softwares, *websites* e sistemas de reconhecimento de sinais baseados em imagens, vídeos ou sensores.

O reconhecimento automatizado de sinais da LIBRAS visa permitir aos computadores a interpretação da língua de sinais reproduzida por seres humanos. O reconhecimento de sinais pode ser aplicado em diversos contextos, como a tradução de conversas entre surdos e pessoas que não compreendem a língua de sinais e a utilização de computadores com interação natural (comandos através de sinais).

Segundo Guimarães et al. [18], a estrutura fonética da língua de sinais brasileira é formada por cinco parâmetros globais: configuração de mão, locação ou ponto de articulação, movimento, orientação da palma da mão e expressão facial. Através da combinação de parâmetros é possível formar sinais assim como, a união de fonemas forma palavras na língua falada. Com base nisso, conclui-se que o reconhecimento dos

parâmetros globais é um ponto de partida para o reconhecimento eficiente de sinais e sentenças da LIBRAS.

O presente trabalho tem como foco contribuir para o reconhecimento automatizado de sinais, utilizando uma abordagem que difere da maior parte dos trabalhos encontrados na literatura; trabalhos que, em geral, utilizam vocabulários restritos ou efetuam o reconhecimento apenas de palavras/sentenças isoladas. Antunes et al. [2] comenta que a LIBRAS não é uma simples soletração da língua falada, portanto o reconhecimento de sinais isolados ou letras do alfabeto (artifícios utilizados pela maioria dos autores) não são suficientes para sua transcrição e interpretação automática. Assim, uma abordagem de reconhecimento deveria basear-se nos elementos mais primitivos da língua: os parâmetros globais.

1.1 Definição do Problema

A configuração de mão é a forma com que a mão da pessoa se encontra durante a execução de determinado sinal, tratando-se pois de um parâmetro primitivo da língua, essencial para o reconhecimento de sinais e sentenças. Segundo Pimenta e Quadros [16] a LIBRAS conta com 61 possíveis configurações de mão apresentadas na Figura 1.1.

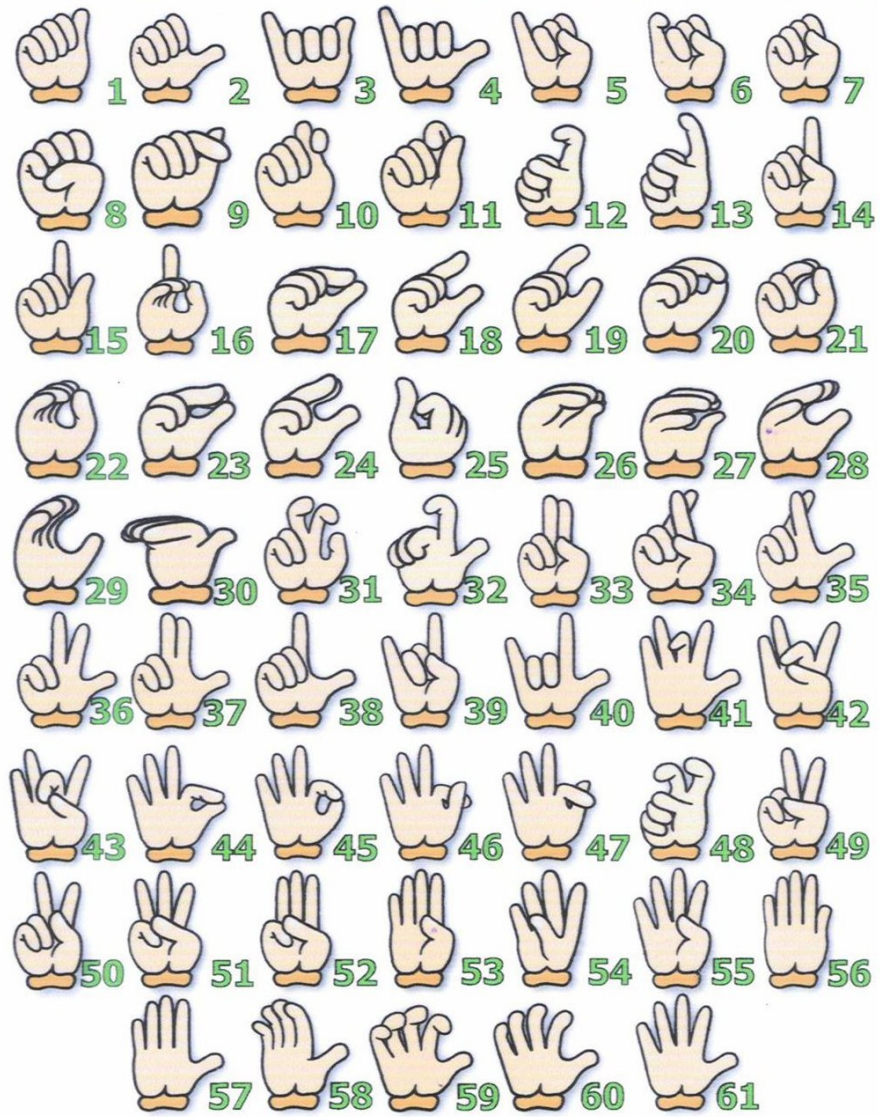


Figura 1.1: Configurações de mão da LIBRAS [16].

Assim, o problema abordado neste trabalho é o reconhecimento das 61 configurações de mão da LIBRAS, parte integrante do problema principal, o reconhecimento da língua de sinais brasileira.

1.2 Objetivos

Baseando-se nos conceitos da abordagem de Antunes et al. [2], este trabalho visa o reconhecimento de um dos parâmetros globais da LIBRAS, a configuração de mão. A abordagem utilizada faz uso de malhas tridimensionais geradas, a partir de imagens das visões frontal e lateral da mão.

Os objetivos específicos deste trabalho são:

1. Aquisição de uma base de dados das configurações de mão da LIBRAS com o sensor Kinect;
2. Reconstrução em 3D da mão sob a forma de malhas tridimensionais;
3. Determinação e extração de características discriminantes das configurações de mão;
4. Classificação das 61 classes correspondentes às malhas tridimensionais das configurações de mão da LIBRAS.

Além disso, o objetivo secundário da geração da base de dados envolve a criação de um conjunto de dados complementar, formado por imagens 2D em visão frontal de cada configuração de mão. A função deste conjunto auxiliar é prover uma imagem 2D correspondente a cada captura de configuração de mão, possibilitando futuras comparações entre métodos de classificação 2D e 3D.

1.3 Justificativa

Conforme citado anteriormente, a maioria dos autores realiza o reconhecimento de sinais isolados, vocabulários restritos ou apenas, algumas sentenças de uma língua. Este tipo de abordagem é útil no contexto de problemas específicos, porém insuficiente ao reconhecimento da língua como um todo.

Destaca-se ainda que, como a LIBRAS não é uma soletração, o reconhecimento de sinais relativos a caracteres não confere o poder de reconhecimento completo da língua. Portanto, a abordagem baseada nos parâmetros globais é mais adequada [2].

Além disso, este trabalho apresenta uma metodologia baseada em malhas 3D e não imagens 2D, como a maioria dos autores utiliza. Uma malha 3D é uma representação em três dimensões de um determinado objeto comumente utilizado em aplicações de realidade virtual e jogos de videogame.

A grande vantagem de se utilizar malhas 3D no reconhecimento de línguas de sinais é a capacidade de identificar detalhes da mão e dedos independentemente da posição e

orientação destes no espaço. Durante a execução de um sinal a mão do ator (aquele que faz o sinal) pode encontrar-se em diversas posições e orientações. A reconstrução da malha 3D possibilita a criação de um objeto invariante a estes parâmetros (corrigindo via software a orientação, posição e escala do objeto 3D ou utilizando um descritor invariante tais parâmetros). Acrescenta-se ainda, que a utilização de modelos 3D evita problemas de oclusão e iluminação como ocorre em imagens bidimensionais.

1.4 Desafios

A maioria dos autores realizou experimentos com bases de dados próprias e utilizou abordagens que não se baseiam nos parâmetros globais [2], implicando na não existência de uma base de dados nas condições necessárias para a realização dos experimentos. A construção da base de dados é o principal desafio na realização deste trabalho.

A geração das malhas 3D das configurações de mão da LIBRAS, utilizando imagens 2D, exige dados de entrada previamente segmentados (apenas a mão deve aparecer na imagem). A segmentação das imagens é realizada de forma facilitada, através da utilização do canal de profundidade do sensor Kinect.

Parte-se do princípio de que a aquisição de malhas 3D, a partir de sequências de vídeo é um problema com solução em desenvolvimento devido ao fato de existirem várias tecnologias sendo criadas para este propósito (sensor LeapMotion, apresentado no capítulo 2). Logo, a ênfase do trabalho será a classificação das malhas e não, sua aquisição.

1.5 Contribuições

As principais contribuições deste trabalho são:

1. Criação de uma base de malhas 3D das configurações de mão da LIBRAS, utilizando a técnica de reconstrução por silhueta;
2. Melhoramento das malhas, através da aplicação de filtros de pré-processamento para suavização e reorganização poligonal;

3. Reconhecimento das configurações, a partir das malhas 3D de mão, utilizando o classificador SVM;
4. Criação de uma base de dados das configurações de mão da LIBRAS com o sensor Kinect (HC-LIBRAS-RGBDS-2011);
5. Criação de uma base de dados complementar com imagens 2D das configurações de mão da LIBRAS (HC-LIBRAS-RGB-2011);

É de fundamental importância ressaltar a contribuição social deste trabalho: o desenvolvimento de sistemas de auxílio a deficientes auditivos contribui significativamente para uma boa utilização dos meios de comunicação digitais por estas pessoas.

1.6 Organização do Trabalho

O trabalho está organizado da seguinte maneira: No capítulo 2 encontram-se os conceitos necessários para o entendimento do trabalho, o capítulo 3 apresenta a revisão bibliográfica, no capítulo 4 é descrito o desenvolvimento do trabalho. O capítulo 5 apresenta os resultados e, por fim, no capítulo 6 encontra-se a conclusão.

CAPÍTULO 2

FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os conceitos necessários para o entendimento do presente trabalho. Entre os assuntos abordados estão: a geração de malhas 3D, o pré-processamento efetuado nas malhas, o descritor *Spherical Harmonics* e o classificador SVM.

Até o momento do desenvolvimento deste trabalho não existe no mercado um sensor capaz de fornecer uma malha 3D da mão com detalhes suficientes para a classificação das configurações de mão. No entanto, existem tecnologias em desenvolvimento que, num futuro próximo, serão capazes de fornecer tais detalhes. Um sensor, ainda em desenvolvimento, porém bastante promissor é o LeapMotion¹ capaz de identificar detalhadamente a mão do usuário (Figura 2.1). Devido a este fato faz-se necessária a criação de uma base de dados artificial para a simulação de experimentos com malhas 3D.

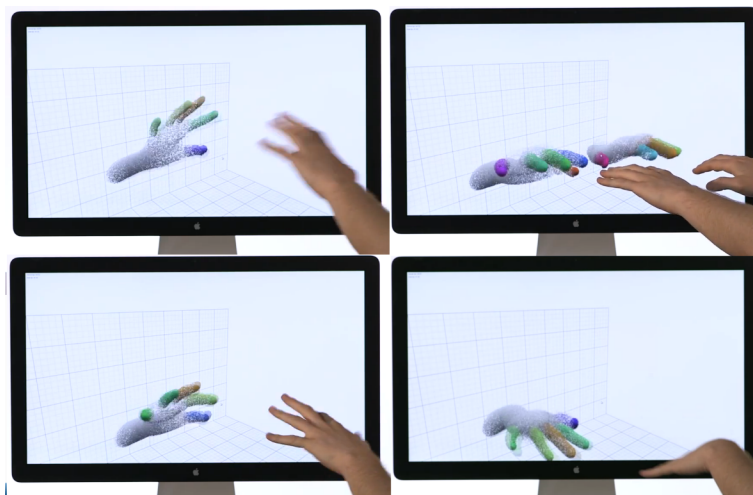


Figura 2.1: Leap Motion.

A utilização de malhas 3D ao invés de imagens 2D fornece benefícios para o reconhecimento das configurações de mão, sendo os principais: evitam-se os problemas de oclusão, malhas 3D são invariantes às condições de iluminação, rotação, translação e

¹<https://leapmotion.com/>

escala².

O capítulo é organizado conforme segue: a seção 2.1 trata da geração de malhas 3D e a técnica utilizada neste trabalho, o descritor de malhas 3D Harmônicos Esféricos (*Spherical Harmonics*) é detalhado na seção 2.2 e, por fim, a seção 2.3 apresenta o classificador utilizado na etapa de testes.

2.1 Geração de Malhas 3D

A reconstrução de objetos em três dimensões é o processo utilizado para gerar malhas 3D, a partir de objetos reais. Existem várias técnicas que podem ser utilizadas neste processo, por exemplo: reconstrução 3D por imagens de visão estéreo [27]; reconstrução com câmeras em movimento [22]; reconstrução 3D em tempo real, utilizando várias câmeras capturando vários ângulos, simultaneamente [6]. O método abordado neste trabalho é a reconstrução por silhueta, utilizado para a geração da base de dados. Esta técnica é apresentada na subseção 2.1.1.

Em alguns casos a reconstrução 3D gera resultados imprecisos, ou seja, malhas com ruídos e imperfeições. Existem técnicas de pré-processamento que otimizam estes resultados, elas são apresentadas na subseção 2.1.2.

2.1.1 Geração da Malha por Silhueta

O método de geração de malhas 3D, a partir de imagens da silhueta do objeto original, é conhecido na literatura como reconstrução por silhueta, outros termos também encontrados são *Visual Hull Construction* e *Shape from Silhouette - SFS*. Este é um método não invasivo (não requer sensores acoplados ao objeto real) utilizado para a criação de malhas 3D, a partir de várias imagens em diferentes ângulos de um determinado objeto [7].

A reconstrução é feita através da estimativa da malha 3D por meio de imagens de dois ou mais pontos de vista de um objeto. Quanto mais imagens forem utilizadas, maior será o nível de detalhe da malha gerada. A Figura 2.2 apresenta um exemplo do método de reconstrução por silhueta aplicado com quatro pontos de vista ao redor do objeto.

²Algumas condições de invariância podem depender do descritor utilizado

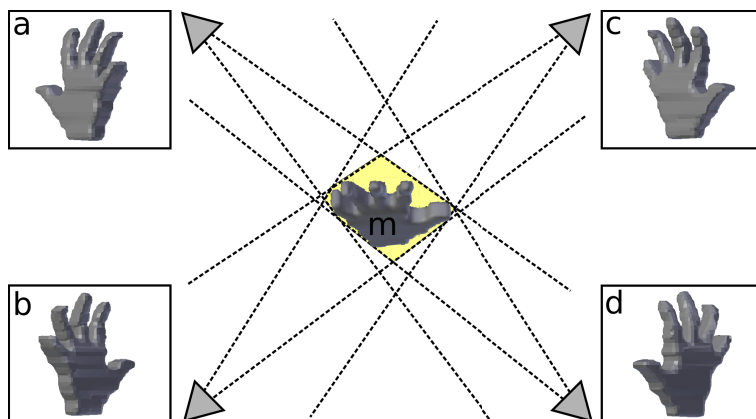


Figura 2.2: Reconstrução por silhueta: Objeto a ser reconstruído (m) e diferentes ângulos de visão (a, b, c, d). Figura inspirada em [7].

Este método gera um volume tridimensional a partir das intersecções de um conjunto de imagens de silhueta de um determinado objeto. Esta técnica exige imagens do objeto previamente segmentadas (apenas a silhueta visível) e coletadas de um objeto estático.

A reconstrução por silhueta utilizada neste trabalho envolve dois conceitos: diferença entre malhas e produção de um volume 3D, a partir de imagens 2D. A diferença entre malhas é uma operação onde dois objetos 3D que se interceptam são utilizados para a criação de uma nova malha cujo formato corresponde à intersecção dos objetos originais. A Figura 2.3 apresenta um exemplo de operação de diferença de malhas, na qual: (a) é a cena de entrada, um cubo e uma esfera que se interceptam, (b) e (c) as visões de cima e frontal da cena em *wireframe*³, (d) representa a malha resultante da diferença entre o cubo e a esfera e, por fim, (e) e (f) são as visões de cima e frontal do resultado da operação em *wireframe*.

³Wireframe é uma técnica de renderização de malhas 3D onde apenas sua estrutura é exibida.

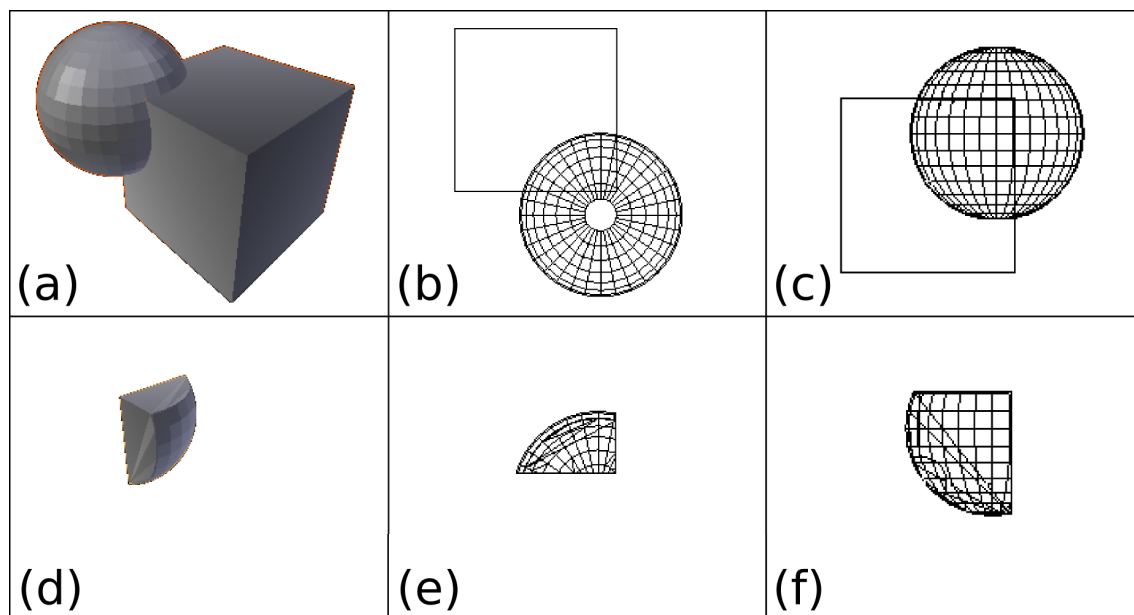


Figura 2.3: Operação diferença entre malhas.

O segundo conceito utilizado no processo de reconstrução 3D por silhueta consiste na utilização de uma imagem 2D para provocar o deslocamento de certos vértices em uma malha 3D de modo formar um contorno tridimensional (silhueta) de um dos lados do objeto. A Figura 2.4 apresenta detalhes da projeção de uma imagem 2D em uma superfície 3D e, como resultado, uma malha com a silhueta da visão frontal da mão.

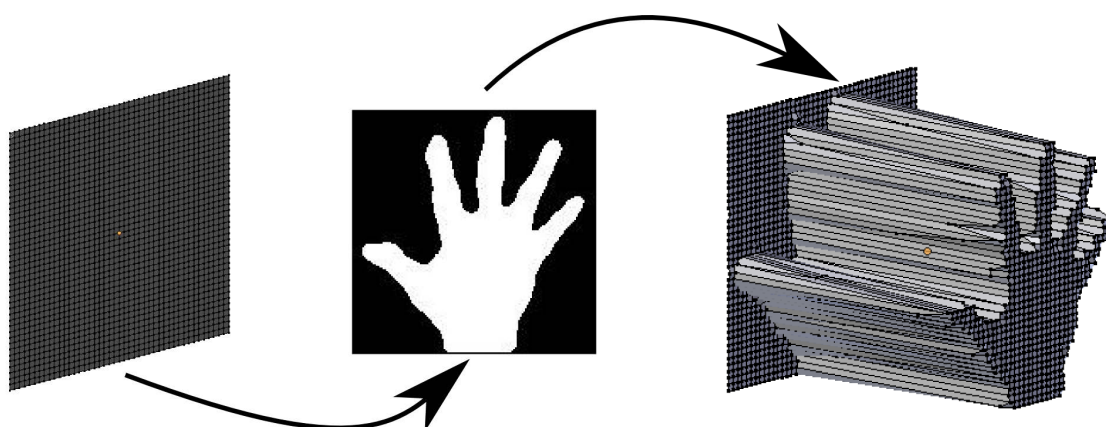


Figura 2.4: Construção da silhueta da visão frontal da mão.

Processando-se a diferença entre as malhas correspondentes às silhuetas das visões frontal e lateral da mão, obtém-se uma malha 3D através do método da silhueta. A Figura 2.5 apresenta um exemplo desta operação, onde (a) são as imagens 2D das visões frontal e lateral da mão, (b) é a cena com as silhuetas e (c) representa a malha final após

a operação diferença.

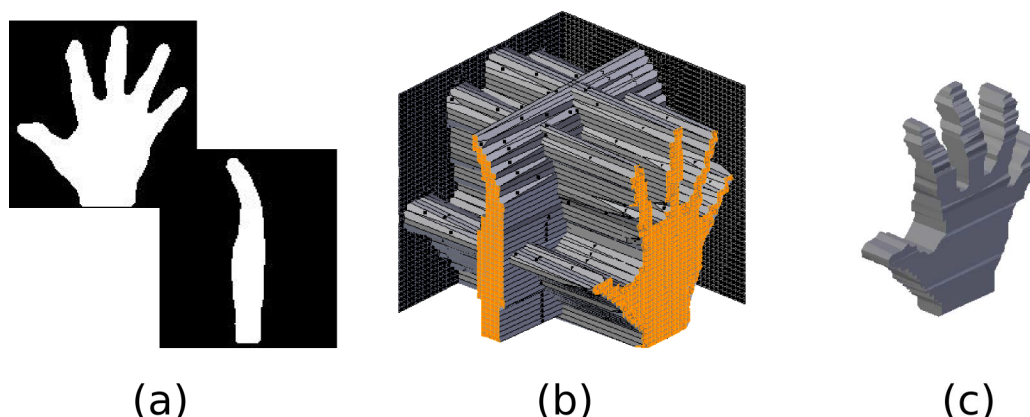


Figura 2.5: Reconstrução da mão pelo método de silhueta usando duas visões.

Utilizando recursos do canal de profundidade do sensor Kinect, a segmentação da mão é feita de forma simplificada se comparada à imagens RGB. Além disso, proporciona imagens no formato de entrada para a criação de versões tridimensionais das configurações de mão, possibilitando a classificação de objetos 3D.

A utilização de malhas 3D na classificação fornece benefícios como: a invariância à condições de iluminação, rotação, translação e escala⁴. Estes recursos são desejáveis no reconhecimento de uma conversação em língua de sinais, pois a mão do ator pode se encontrar em inúmeras posições e orientações durante a execução de um sinal.

Uma desvantagem do método da silhueta é que determinados detalhes (como concavidades internas) são impossíveis de serem recriados independentemente da quantidade de imagens utilizada [37].

2.1.2 Pré-Processamento das Malhas

A reconstrução por silhueta com duas imagens resulta em malhas com ruídos, a aplicação de filtros de suavização visa melhorar a qualidade destas malhas. É considerada uma melhoria de qualidade o fato de a malha suavizada ser visualmente mais semelhante à mão real do que a malha não suavizada. Como não existe sensor capaz de fornecer malhas 3D nas condições necessárias para realização deste trabalho, é de grande importância a

⁴Alguns parâmetros são dependentes do descritor utilizado.

semelhança entre as malhas artificiais e à mão real, de forma simular o resultado obtido com um sensor ideal.

Utilizando-se algoritmos de remapeamento 3D, é possível reorganizar a malha de modo que o mesmo objeto possa ser criado com uma estrutura tridimensional mais organizada e suave. A reconstrução por silhueta utilizada neste trabalho gera malhas com polígonos significativamente irregulares como mostra a Figura 2.6a, o algoritmo de remapeamento 3D utilizado neste trabalho é o Dual Contouring (detalhado por Ju [23]), que converte a estrutura da malha em polígonos mais padronizados, deixando a topologia 3D da malha mais organizada como é possível notar na Figura 2.6b.

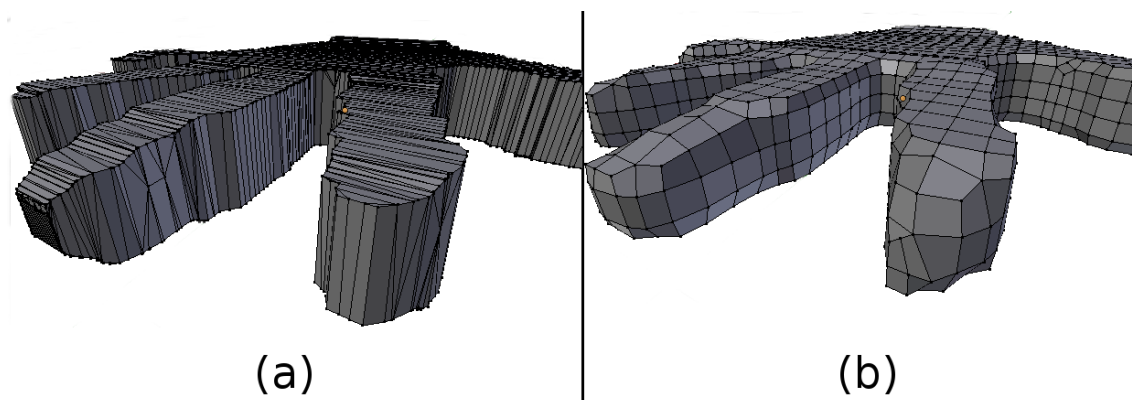


Figura 2.6: Reorganização poligonal, malha original (a) e malha reorganizada (b).

Além disto, a aplicação deste algoritmo suaviza a malha, eliminando pequenos ruídos no objeto 3D e deixando-a mais parecida com o objeto real. Em uma base de dados com malhas geradas sinteticamente, como neste trabalho, é muito importante que os objetos virtuais sejam semelhantes à mão real. Quanto mais semelhança houver entre o objeto virtual e o original mais válido será o experimento.

O *software* Blender 3D⁵ possui recursos para suavizar malhas tridimensionais e pode ser operado automaticamente por *scripts* na linguagem python. O filtro de reorganização poligonal⁶ é aplicado conforme demonstra a Figura 2.7, onde (a) é a malha resultante da reconstrução por silhueta e (b) é a malha suavizada.

⁵<http://www.blender.org>

⁶Algoritmo nativo do Blender 3D, implementação baseada em Ju [23].

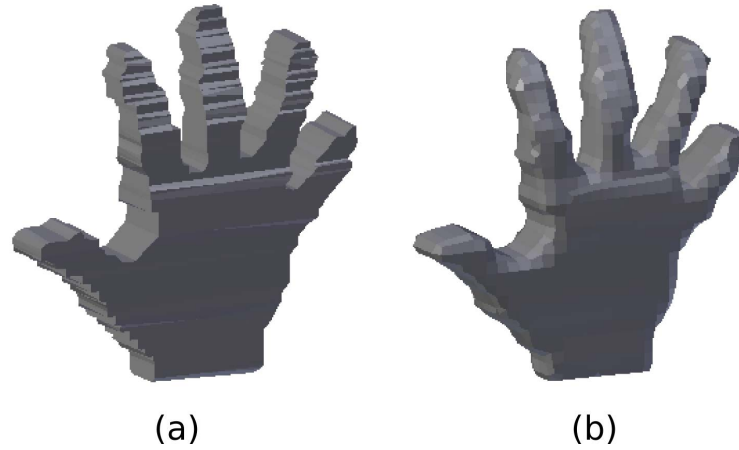


Figura 2.7: Suavização da malha.

2.2 Harmônicos Esféricos (Spherical Harmonics)

A comparação de malhas tridimensionais é um método que permite a classificação de objetos e fornece recursos interessantes como invariância à condições de iluminação, translação, rotação e escala. *Spherical Harmonics* é um método descritor de malhas 3D que atende à estes requisitos e pode ser aplicado neste caso.

Um método de representação de malhas 3D por *Spherical Harmonics* invariante à rotação foi descrito e apresentado por Kazhdan [25]. O autor disponibiliza uma ferramenta⁷ com a implementação do método, que recebe como entrada malhas 3D em formato Sanford (.ply) e retorna como saída um arquivo binário com a assinatura da malha.

O cálculo do descritor é realizado da seguinte maneira:

1. Dado um modelo $\mathbf{M} \subset \mathbb{R}^3$, transladar sua malha de polígonos de modo que o centro de massa do modelo esteja na origem do sistema de coordenadas.
2. Reescalar a malha de modo que a raiz quadrada da distancia euclidiana média do centro de massa até os vértices dos polígonos do modelo seja 0.25. Isto garante, com alta probabilidade, que todos os vértices $\mathbf{v} \in \mathbf{M}$ estarão contidos em uma esfera de raio 0.5 ao redor da origem.
3. Rasterizar os polígonos criando uma imagem binária 3D, ou *voxel grid*, $I \subset \mathbb{Z}^3$

⁷<http://www.cs.jhu.edu/~misha/HarmonicSignatures/>

de dimensão 64^3 (atribuindo o valor 1 aos *voxels* que estejam a até um voxel de distância de um ponto do polígono, e 0 para os outros *voxels*).

O processo de rasterização pode ser feito da seguinte forma: seja $i(\mathbf{v}) = 32 \cdot \mathbf{v} + (32, 32, 32)^T$, então

$$I(i(\mathbf{v})) = 1, \forall \mathbf{v} \in \mathbf{M}$$

garantindo que todos os vértices da malha estarão contidos dentro do *grid* I .

4. Calcular a *transformada de distância euclidiana* ($EDT(p)$) de cada ponto $p \in I$.

Exponenciar a distância para obter:

$$I_{ed}(p) = e^{-EDT(p)^2/16}, \forall p \in I. \quad (2.1)$$

A exponenciação atribui aos *voxels* que coincidem com a fronteira do objeto (bordas) valores próximos a 1.0, decrescendo exponencialmente à medida em que aumenta a distância.

5. O *voxel grid* então é tratado como uma função definida no espaço \mathbb{R}^3 , e restrito por um conjunto de esferas concêntricas de raio $r = \{1, 2, \dots, 32\}$. Cada uma das 32 restrições esféricas é dada por uma função esférica ($f^r(\theta, \phi)$), calculada por decomposição em esféricos harmônicos, e amostrada em uma grade de longitude (θ) e latitude (ϕ) de dimensão 64^2 :

$$f^r(\theta, \phi) = \sum_{l=1}^{\infty} \sum_{m=-l}^l a_{lm} Y_l^m(\theta, \phi) \quad (2.2)$$

6. Calcular a energia de cada função $f^r(\theta, \phi)$, representada pela norma L_2 ($L2-Norm$) dos 16 componentes de mais baixa frequência, ou seja, os primeiros 16 componentes harmônicos:

$$SH(f^r) = \{\|f_0^r(\theta\phi)\|, \|f_1^r(\theta\phi)\|, \dots, \|f_{16}^r(\theta\phi)\|\} \quad (2.3)$$

onde f_l^r é o componente harmônico l de f^r , escalado de acordo com o raio:

$$f_l^r(\theta, \phi)^r = \frac{1}{\sqrt{4\pi r^2}} \left(\sum_{m=-l}^l a_{lm} Y_l^m(\theta, \phi) \right) \quad (2.4)$$

7. Sabendo que rotações não afetam a norma dos componentes harmônicos ($\|f_l^r\|$), define-se $SH(f^r)$ (Equação 2.3) como a assinatura da função f^r .
8. Finalmente, as assinaturas das 32 funções são combinadas em uma matriz bidimensional em que o valor do índice (i, j) é igual à norma do componente de ordem j da função esférica da esfera de raio i (f_j^i).

O autor comenta que o método possui uma limitação quanto ao alinhamento das malhas 3D, requisito necessário para que o descritor seja invariante à rotação em todos os casos. A solução do problema é dada com o uso de valores de alinhamento independentes para cada frequência, devido a isto, a matriz do descritor conta com uma linha adicional, responsável por armazenar estes valores. Assim, a ferramenta de cálculo do Spherical Harmonics de Kazhdan gera uma matriz $\mathbf{A} \subset \mathbb{R}^2$, de dimensões 17×32 , cujos elementos correspondem à 32 assinaturas $SH(f^r)$ unidas a 32 parâmetros de alinhamento. O autor não divulga quais posições da matriz correspondem aos valores de alinhamento, e realiza a comparação de assinatura utilizando todos os valores, sugerindo que os mesmos devem ser utilizados em conjunto com o restante da matriz (subconjunto 16×32 do cálculo original) para a formação do vetor de características.

A Figura 2.8 ilustra o cálculo do descritor, onde: dado um objeto 3D \mathbf{M} , é feita a rasterização da malha em um *voxel grid* 64^3 (1), em seguida o *grid* é decomposto em 32 funções esféricas $f^r(\theta, \phi)$ (2), cada função é então decomposta como a soma dos 16 primeiros componentes harmônicos, análogo à decomposição de Fourier (3). Cada uma das 16 decomposições gera uma assinatura (4), por fim a combinação de todas elas gera a assinatura da malha 3D (5)[24].

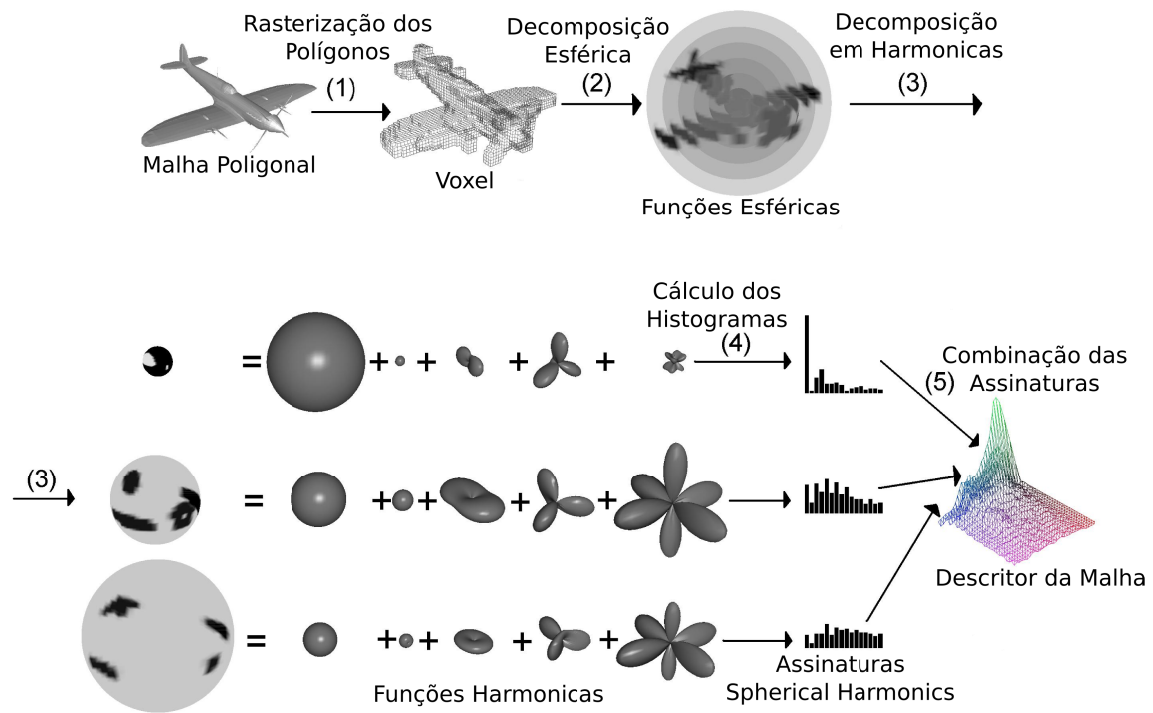


Figura 2.8: Ilustração do cálculo do descritor Spherical Harmonics[24].

O autor representa a assinatura da malha por meio de histogramas bidimensionais armazenados sob a forma de matriz. A comparação entre duas assinaturas pode ser feita utilizando-se distância euclidiana entre os elementos de duas matrizes, sendo que o valor zero indica que as duas assinaturas correspondem à mesma malha e, quanto maior o valor obtido, maior é a diferença estrutural entre elas. As Figuras 2.9(a) e 2.9(b) apresentam exemplos de histogramas gerados por este método.

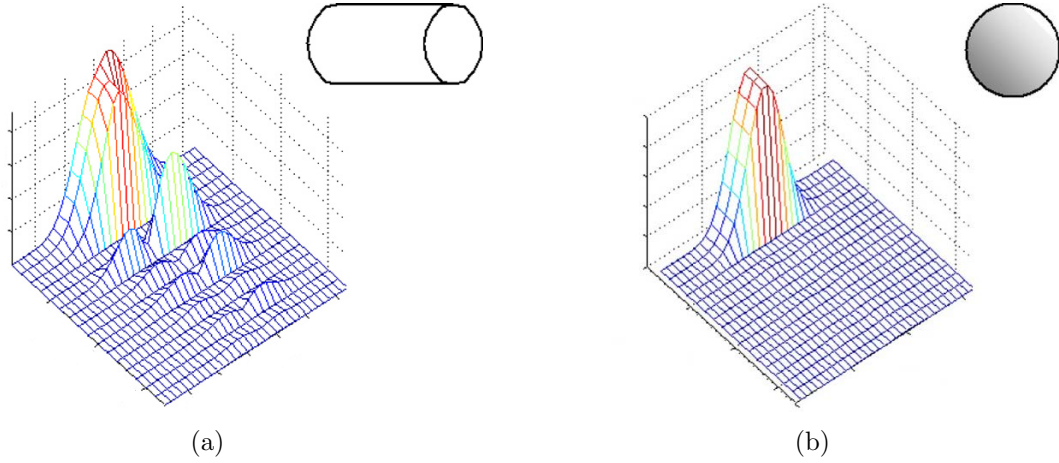


Figura 2.9: Histogramas das assinaturas geradas por *Spherical Harmonics* a partir de duas malhas: Cilindro (a) e Esfera (b) (baseada em [8]).

O arquivo de assinatura gerado pela ferramenta de Kazhdan é composto por dois inteiros positivos, os quais representam as dimensões matriz que forma o histograma, seguido de um lista de $X \times Y$ números em ponto flutuante que compõe a assinatura da malha 3D, onde X e Y são as dimensões da matriz. O autor demonstra a eficiência do descritor e da ferramenta, utilizando uma base com 1890 malhas 3D divididas em 85 classes. O descritor foi comparado à outros seis métodos utilizando as métricas *precision* e *recall*, a performance do *Spherical Harmonics* obteve resultados entre 46% e 245% acima dos valores obtidos pelos outros descritores [24].

Um comparativo entre descritores de malhas e classificação manual foi realizado por Clark et al. [8]. O objetivo do trabalho foi mostrar que, dada uma malha 3D, os descritores são capazes de fornecer meios de identificação de malhas similares com eficiência similar ao mesmo processo realizado por humanos. Foram comparados três descritores: *D2 shape distribution*, *Spherical Harmonics* e *Surface Partitioning Spectrum*. Segundo o autor, todos os descritores obtiveram boa qualificação nos testes e *Spherical Harmonics* demonstrou-se o método mais eficaz devido à baixa taxa de falsos-negativos quando comparado aos demais descritores.

Com base nos dados fornecidos por Kazhdan [24] e o comparativo de Clark et al. [8], conclui-se que o descritor *Spherical Harmonics* é um método eficaz e possui poder descritivo suficiente para a formação do vetor de características das malhas 3D utilizadas

neste trabalho.

2.3 Support Vector Machine

Support Vector Machine (SVM), é um classificador que faz uso de aprendizado supervisionado e tem como principal objetivo a classificação em problemas com duas classes [9]. De modo geral o SVM pode ser aplicado em conjuntos de dados não separáveis linearmente e, além disso, é utilizável em bases de dados com múltiplas classes através do uso de abordagens como um-contra-todos [36] e um-contra-um ou *pairwise* [1] [4].

O SVM foi proposto por Vapnik em 1998 [48], o método pode ser considerado relativamente novo se comparado à maioria dos classificadores mais utilizados na literatura. Este classificador trabalha com a separação de classes através do uso de hiperplanos, que são sub-espacos lineares. Um hiperplano em um espaço linear de dimensão n é um plano com dimensão $n-1$. A Figura 2.10 apresenta um exemplo de hiperplano separando duas classes em um espaço bidimensional.

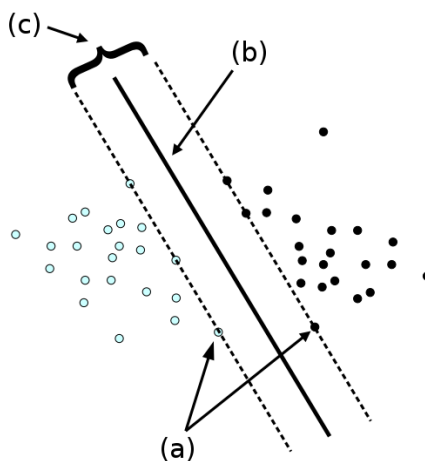


Figura 2.10: Hiperplano separando duas classes, onde (a) são os vetores de suporte, (b) é o hiperplano e (c) são as margens (distâncias entre as classes e o hiperplano) representadas pelos pontos escuros e claros [35].

O treinamento do SVM consiste do cálculo do melhor hiperplano para a separação das classes, onde a distância entre as classes e o hiperplano (margens) é maximizada. O problema apontado na Figura 2.11a pode possuir infinitas soluções, alguns exemplos estão representados por linhas pontilhadas na Figura 2.11b, a solução ótima é formada

pelo hiperplano que maximiza as margens 2.11c.

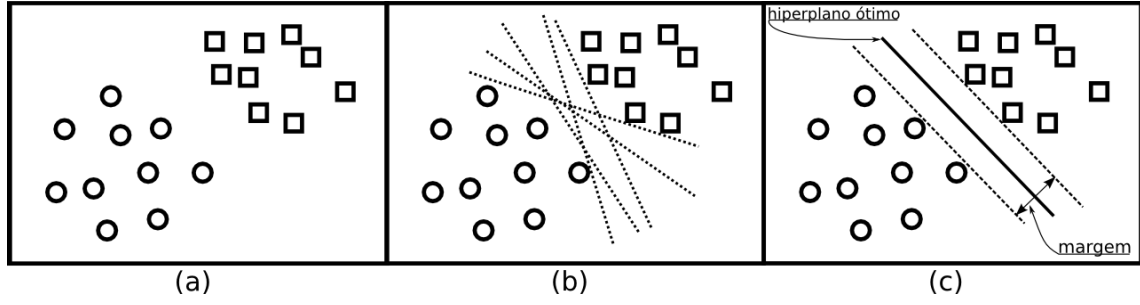


Figura 2.11: Problema com duas classes (a), exemplos de hiperplanos soluções (b) e hiperplano ótimo (c).

A separação demonstrada na Figura 2.10 demonstra um conjunto de dados linearmente separável, grande parte dos problemas de classificação não podem ser resolvidos desta forma, devido à possuírem conjuntos de dados não separáveis linearmente. Uma maneira de se trabalhar com estes conjuntos de dados é através da utilização de SVMs não lineares.

Segundo Ormonde [38], pode-se descrever o funcionamento de um SVM linear da seguinte maneira:

Dado um conjunto de dados de treinamento CT , composto por n instâncias da forma:

$$CT = (\vec{x}_i, y_i) | \vec{x}_i \in \mathbb{R}, y_i \in \{-1, 1\}, \quad i = 1, \dots, p \quad (2.5)$$

na qual y_i é 1 ou -1, indicando a classe a qual a instância x_i pertence. \vec{x}_i é um vetor p -dimensional de números reais. O conjunto é dito linearmente separável se existir um vetor \vec{w} e um escalar b tal que as restrições apontadas na Equação 2.6 sejam válidas para todo o conjunto CT .

$$\begin{cases} \vec{w} \cdot \vec{x}_i + b \geq 1, & \text{se } y_i = 1 \\ \vec{w} \cdot \vec{x}_i + b \leq -1, & \text{se } y_i = -1 \end{cases} \quad (2.6)$$

Sendo o conjunto CT linearmente separável, é possível selecionar dois hiperplanos h_1 e h_2 capazes de representar as fronteiras dos conjuntos de dados (vetores de suporte na Figura 2.10), eles são definidos por:

$$\begin{cases} h_1 = \vec{w} \cdot \vec{x}_i + b = 1 \\ h_2 = \vec{w} \cdot \vec{x}_i + b = -1 \end{cases} \quad (2.7)$$

Dados os hiperplanos h_1 e h_2 paralelos, o hiperplano ótimo é o que maximiza a distância entre h_1 e h_2 e, conseqüentemente, maximiza a distância entre as classes -1 e 1. Esta distância é dada por:

$$\frac{2}{\|\vec{w}\|} \quad (2.8)$$

Com isso, durante o processo de treinamento do SVM procura-se a minimização de $\|\vec{w}\|^2$ sujeito à restrição: $y_i(\vec{w} \cdot \vec{x}_i - b) \geq 1$, $i = 1, \dots, n$. O problema então pode ser reescrito utilizando os multiplicadores de Lagrange [3] $\{a_i\}_{i=1}^p$ na restrição da inequação anterior, obtendo-se o seguinte problema de minimização:

$$L_P = \frac{1}{2} \|\vec{w}\|^2 - \sum_{i=1}^p a_i y_i (\vec{x}_i \cdot \vec{w} + b) + \sum_{i=1}^p a_i, \quad a_i \geq 0 \quad (2.9)$$

Este problema pode ser convertido no seguinte problema dual de maximização:

$$L_D = \sum_{i=1}^p a_i - \frac{1}{2} \sum_{i=1}^p \sum_{j=1}^p a_i a_j y_i y_j (\vec{x}_i \cdot \vec{x}_j) \quad (2.10)$$

sujeito a:

$$\begin{cases} \sum_{i=1}^p a_i y_i = 0 \\ a_i \geq 0 \end{cases} \quad (2.11)$$

A solução do problema é um vetor de coeficientes $\vec{a}^T = (a_1, \dots, a_p)^T$ para os quais o problema de maximização anterior apresenta valor máximo. Dados os coeficientes \vec{a}^T é possível calcular o hiperplano ótimo através da Equação 2.12 [10]:

$$\vec{w} = \sum_{i=1}^p a_i y_i \vec{x}_i \quad (2.12)$$

Dado um vetor de suporte \vec{x}_{vs} cujos coeficientes a_i sejam diferentes de zero, o valor do

escalar b é dado substituindo-se \vec{x}_{vs} na Equação 2.13:

$$b = y_{vs} - \vec{w} \cdot \vec{x} \quad (2.13)$$

Calculados o vetor \vec{w} e o escalar b , o SVM está apto a realizar a classificação de novas instâncias. A classificação é feita de acordo com a seguinte função de decisão $f(\vec{x}) = \vec{w} \cdot \vec{x} + b$, se $f(\vec{x}) > 0$ $\vec{x} \in$ classe A, caso contrário $\vec{x} \in$ classe B.

O modelo tradicional de SVM é aplicável a problemas com duas classes, a seguir são apresentados detalhes das estratégias aplicáveis a problemas com multiplas classes.

A estratégia um-contra-um ou *pairwise* consiste na construção de um SVM para cada par de classes. Assim, para um problema com c classes serão treinados $c(c-1)/2$ SVMs para a distinção de todas as classes [36]. Dadas as saídas de todos os SVMs podem ser utilizadas várias estratégias para classificação de uma amostra, dentre elas Price et al. [41] e Hastie e Tibshirani [19].

Outra estratégia utilizável em problemas com multiplas classes é a um-contra-todos, onde para cada classe é treinado um SVM, responsável por distinguir exemplos de uma classe de exemplos das classes restantes. Em seguida, as saídas de todos os SVMs são mapeadas em probabilidades, utilizadas para classificar a amostra em uma das possíveis classes [36]. Um método para estimativa das probabilidades em SVMs um-contra-todos pode ser observado no trabalho de Platt [40].

CAPÍTULO 3

REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta uma revisão dos trabalhos mais relevantes na área do reconhecimento de línguas de sinais. A discussão dos trabalhos está distribuída ao longo do texto. Além disso, ao final do capítulo é apresentada uma revisão das principais bases de dados focadas no reconhecimento de sinais.

O reconhecimento de línguas de sinais é um tema bastante estudado na literatura. Segundo Quan [42] as pesquisas na área de reconhecimento de línguas de sinais começaram em 1982 com o trabalho de Shantz [44].

A literatura apresenta duas principais abordagens utilizadas no reconhecimento de línguas de sinais: abordagem baseada em visualização e abordagem baseada em luvas sensoriais [42] [51]. A seção 3.1 apresenta detalhes destas abordagens e alguns dos principais trabalhos realizados no reconhecimento de línguas de sinais, em seguida, a seção 3.2 exibe um resumo geral de todos os trabalhos. Por fim, a seção 3.3 apresenta as principais bases de dados encontradas na literatura.

3.1 Reconhecimento de Línguas de Sinais

As abordagens utilizadas no reconhecimento de línguas de sinais podem ser divididas da seguinte maneira: A primeira separação ocorre no modo de percepção, visual ou sensorial. Nos sistemas visuais as informações são obtidas através da captura, segmentação e extração de características a partir de imagens e/ou sequências de vídeo do gesticulador. Sistemas sensoriais consistem na utilização de luvas ou acessórios com sensores¹ capazes de capturar dados da mão e dos movimentos gerados pelo usuário no momento da execução de um gesto.

Estas abordagens podem novamente ser divididas em estáticas (como imagens ou dados

¹São considerados sensores os acessórios de medição de dados acoplados ao usuário como acelerômetros e braços robóticos.

de sensores capturados em um determinado momento) e dinâmicas (sequências de vídeo ou fluxos de dados de sensores).

3.1.1 Abordagem baseada em visualização

Esta seção tem como objetivo apresentar os principais trabalhos desenvolvidos com base na abordagem de visualização. Primeiramente são apresentados os trabalhos que usam a abordagem estática, ou seja, aqueles em que as bases de dados são compostas por fotos ou dados isolados.

Marcotti [30] construiu uma base de dados com quatro conjuntos de imagens cada um contendo 21 fotos de diferentes sinais da LIBRAS. As características extraídas das imagens foram a posição relativa da mão no plano na imagem, a área relativa do objeto (área da mão fazendo o gesto em relação à “área útil” segmentada) e o momento de inércia em torno dos eixos horizontal e vertical. A classificação foi baseada em uma árvore de decisão gerada com o algoritmo J48 do software Weka. Não constam dados das taxas de acertos do algoritmo.

O reconhecimento de sinais com a abordagem visual em geral depende de um bom posicionamento do gesticulador frente à câmera. Assim Wang [52] propõe uma alternativa baseada em captura estéreo de imagens. Foram utilizadas duas câmeras capturando diferentes ângulos da cena e, com base nas informações de duas câmeras o sistema estéreo gera um novo ponto de visão alinhado conforme mostra a figura 3.1. As três imagens servem de entrada para o sistema de reconhecimento.

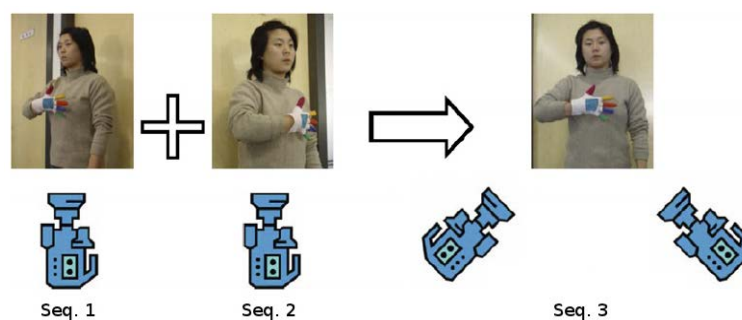


Figura 3.1: Sistema de visão estéreo utilizado por Wang [52].

Para facilitar a segmentação da mão e dos dedos o autor utiliza luvas coloridas no

processo. A base de dados foi capturada pelo autor e o vocabulário utilizado foi composto por 100 signos da língua de sinais chinesa, o classificador foi o HMM (*Hidden Markov Models*), as matrizes fundamentais do sistema de visão estéreo foram utilizadas como características (Detalhes na seção 4.3 do artigo de Wang [52]) e as taxas de reconhecimento chegaram a 94%.

Starner [45] desenvolveu um sistema para reconhecimento de sinais da língua americana (ASL). O sistema conta com dois modos de operação, onde a variação acontece no posicionamento da câmera com relação ao ator. Existem dois posicionamentos: de frente para o ator e fixada no boné apontando para baixo (focando nas mãos).

A base de dados utilizada por Starner [45] foi gravada pelo próprio autor. Ela foi criada a partir de um vocabulário composto por 40 palavras divididas entre verbos, pronomes, substantivos e adjetivos do qual se extraíam combinações para formar sentenças. Dentre as características utilizadas para a classificação estão: a posição XY da mão na imagem, a translação da mão no decorrer dos quadros e o tamanho da área correspondente à mão na imagem. O classificador utilizado foi o HMM e os resultados do experimento com a câmera na frente do usuário apresentaram cerca de 92% acerto, já o experimento com a câmera no boné obteve 97% de amostras classificadas corretamente. Apesar de gerar boas taxas de reconhecimento o sistema da câmera no boné proposto por Starner [45] possui limitações, como a impossibilidade de identificação da expressão facial e a falta de praticidade do sistema gerada pelo uso do boné.

Quan [42] descreveu experimentos que consistem da classificação de sinais com base em informações espaciais e temporais extraídas de sequências de vídeos. A base de dados utilizada nos experimentos foi capturada pelo próprio autor e consistia de 30 letras do alfabeto manual chinês com 195 imagens representando cada letra totalizando 5850 imagens. O autor utiliza cinco características para discriminar as imagens: o histograma de cores, 7 momentos de Hu, 48 Filtros de Gabor e 128 descritores de Fourier. O classificador utilizado por Quan [42] foi o SVM (*Support Vector Machines*) e a média das taxas de acerto dos experimentos foi 95.55%.

Li [29] desenvolveu um sistema de reconhecimento de sentenças da língua de sinais

americana utilizando o sensor Kinect. O vetor de características foi baseado nas junções do corpo fornecidas pelo Kinect e o autor utilizou *template matching* como técnica de comparação dos sinais. Não foram reportadas taxas de acerto, o autor apenas comenta que o protótipo é capaz de reconhecer com precisão uma lista de 11 frases.

Keskin [26] realizou o reconhecimento das configurações de mão da língua de sinais americana referentes aos 10 dígitos utilizando o Kinect. O autor utilizou uma base sintética (imagens renderizadas a partir de um modelo) para treinamento e testes e, posteriormente aplicou o sistema em ambiente real (utilizando imagens do Kinect). O método se baseia na obtenção de um esqueleto 3D da mão (Figura 3.2a), o qual juntamente com 21 partes segmentadas da mão (Figura 3.2a) forma o vetor de características. Os classificadores utilizados no experimento foram Florestas de Decisão (*Decision Forests*), Redes Neurais e SVM, em todos os casos o SVM obteve melhor resultado, sendo que o melhor caso apresentou uma taxa de acerto de 99.9%.

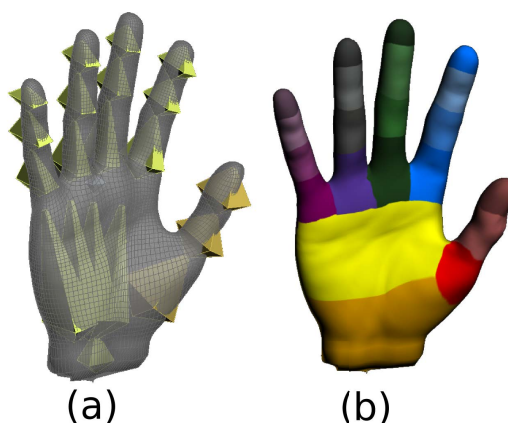


Figura 3.2: Esqueleto 3D (a) e mão segmentada (b) do método de Keskin [26].

3.1.2 Abordagem baseada em sensores

Mehdi [34] utilizou uma luva com sensores para obter dados da configuração de mão de modo estático. Foram utilizados informações de 7 sensores para formar o vetor de características, um deles calcula a inclinação da mão, outro calcula a rotação e 5 deles capturam informações dos dedos. Uma leitura completa dos 7 sensores resulta em dados que representam a configuração de mão em um determinado instante de tempo.

A base de dados utilizada por Mehdi [34] foi capturada pelo próprio autor e foi com-

posta por um vocabulário formado por 24 das 26 letras do alfabeto de sinais americano, Duas letras foram removidas pois utilizavam movimento e, devido a isto, não poderiam ser utilizadas no sistema implementado. Foi utilizada uma rede neural para a classificação dos sinais e a taxa de acerto foi de 88%.

O trabalho de Wang [51] trata do reconhecimento da língua de sinais americana (*American Sign Language* - ASL). Foi utilizada uma luva com 18 sensores lendo informações da mão e dos dedos. Os sensores da luva coletam informações e um rastreador eletrônico calcula as posições 3D de cada um deles. A base de dados foi capturada pelo autor. Utilizando a técnica HMM multi-dimensional e características obtidas a partir das posições dos sensores o autor treina e testa o sistema com um conjunto de 26 letras do alfabeto e 36 configurações de mão. As taxas de acerto do sistema variaram entre 90% e 97%.

A detecção automática de sinais tem sido utilizada em vários contextos tais como tradutores e dicionários. Zafrulla [54] desenvolveu um sistema de verificação de frases para um jogo eletrônico focado no aprendizado de crianças surdas chamado CopyCat. Foram capturadas 1204 frases de 11 usuários. Foram utilizadas duas luvas: uma vermelha e uma roxa, cada uma com um acelerômetro capaz de capturar os movimentos gerados na reprodução das frases. O classificador utilizado foi o HMM e o vetor de características foi formado por uma combinação de atributos extraídos da visão de uma câmera e dos acelerômetros das luvas. Segundo o autor a taxa de acerto para a verificação das frases foi de 83%.

Posteriormente, o mesmo autor desenvolveu uma solução para o mesmo problema utilizando o sensor Kinect. Os experimentos foram realizados em duas tomadas, na primeira os usuários estavam sentados e na segunda em pé. O classificador utilizado foi o HMM e o vetor de características foi baseado na posição das junções do corpo fornecidas pelo Kinect. As taxas de acerto na verificação das frases foram de 51.5% com os usuários sentados e 76.12% em pé [53].

Wang [50] trabalhou no reconhecimento contínuo baseado em fonemas da língua de sinais chinesa. Foram utilizadas duas luvas com 18 sensores cada e um aparelho de

rastreamento 3D para a captura das configurações, posicionamento e orientações das mãos. A base utilizada nos experimentos foi capturada pelo autor e consiste de 5119 fonemas e, através da união de 2 ou mais fonemas, o autor formou 200 sentenças. O classificador utilizado foi o HMM e trabalhou tanto com fonemas isolados quanto com sentenças. O vetor de características foi formado pelos contornos, posições e orientações das mãos. As taxas de reconhecimento para fonemas isolados atingiram a 100% no melhor caso, já a detecção dos fonemas em sequencias contínuas de sinais foi de 92.8%.

Vogler [49] apresenta estudos e experimentos baseados em signos estáticos obtidos através de um aparelho de rastreamento 3D (*Flock of Birds*). Ele propôs uma abordagem baseada no reconhecimento da língua de sinais americana em fluxos contínuos de dados, trabalhando com parâmetros tridimensionais obtidos dos sensores. Utilizando os dados do aparelho ele reconstruiu em 3D a estrutura dos braços do gesticulador conforme mostra a figura 3.3.

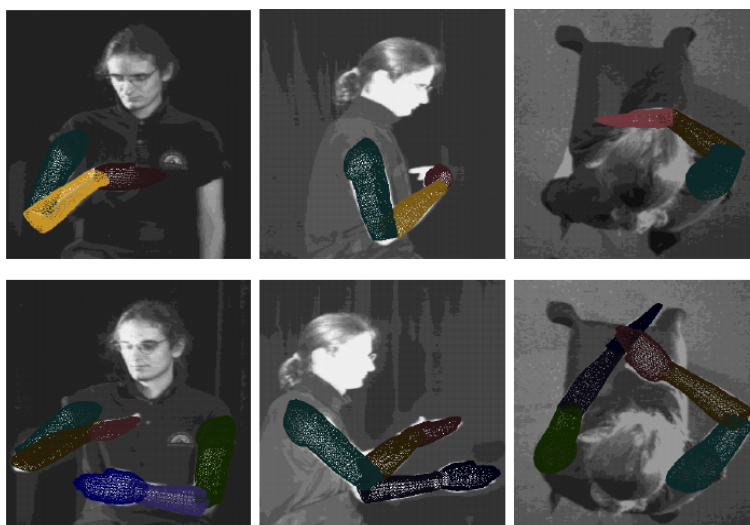


Figura 3.3: Reconstrução 3D dos braços utilizado por Vogler [49].

A base de dados foi capturada pelo autor. O vocabulário utilizado consiste de 53 signos, não listados pelo autor. As características utilizadas foram as posições 3D dos braços do usuário em diversos quadros. O classificador utilizado foi uma variação do método HMM e as taxas de reconhecimento dos experimentos atingiram 100% de acerto no melhor caso.

O autor alega que as informações do braço do gesticulador são discriminantes para

o reconhecimento dos gestos por ele utilizados, logo todas as informações da mão são descartadas, utilizando-se apenas dados dos braços no processo de reconhecimento.

Também foram encontrados alguns desafios no processo de reconhecimento, Zafrulla [54] por exemplo comenta da dificuldade gerada pelas variações nos sinais, usuários diferentes nem sempre fazem o mesmo sinal da mesma maneira. Variações podem ser causadas por inúmeros fatores tais como localização regional (analogamente pode ser comparado ao sotaque, regiões diferentes podem gerar signos diferentes) e até diferenças anatômicas (diferenças anatômicas podem causar distorções na reprodução de determinadas configurações de mão, A Figura 3.4 exemplifica diferenças anatômicas de uma mesma configuração de mão descritas por Flasiński [31]).

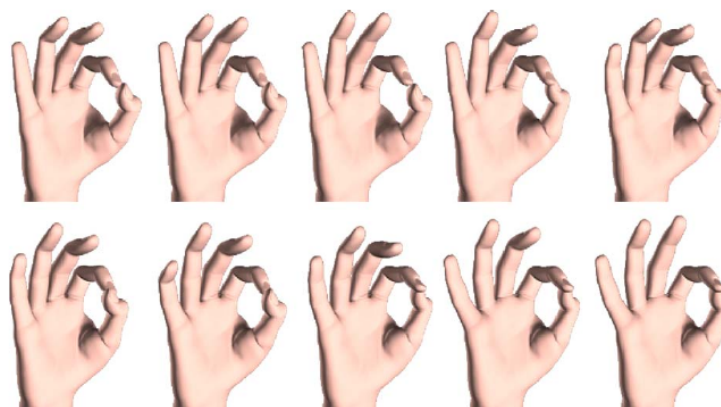


Figura 3.4: Distorções geradas por diferenças anatômicas de uma mesma configuração de mão [31].

Através da análise dos principais trabalhos na área de reconhecimento de línguas de sinais foi possível perceber que, na maioria dos casos, os sistemas fazem uso de artifícios como luvas ou vocabulários controlados, isto dificulta sua utilização em ambientes reais. Luvas e sensores são incômodos ao usuário e, muitas vezes, limitam a execução de movimentos e a captura de outros parâmetros essenciais das línguas de sinais (como a expressão facial por exemplo). Entretanto, a abordagem de sensores (em especial Wang [50]) indica que com uma boa detecção da mão é possível obter uma boa taxa de reconhecimento.

Também é importante ressaltar que sistemas baseados em imagens são suscetíveis a problemas de iluminação, oclusão entre outros. Assim, sensores 3D que consigam capturar a mão por completo podem fornecer a precisão das luvas e a comodidade da abordagem

visual.

Acrescenta-se ainda que a abordagem utilizada pela maioria dos autores é focada no reconhecimento de palavras isoladas, letras ou números. Segundo Peres et. al. [39] e Antunes et al. [2] línguas de sinais são sistemas linguísticos complexos que possuem regras e estruturas como fonologia, morfologia, sintaxe, semântica, logo um sistema eficiente precisaria ser treinado para reconhecer todas as palavras de uma determinada língua. Segundo Dreuw e Ney [15] e Dreuw et al. [14] o uso de vocabulários com palavras isoladas para o reconhecimento de línguas de sinais é inadequado devido à grande quantidade de dados necessária para o treinamento do sistema.

Conclui-se então que para a construção de um sistema que seja capaz de reconhecer LIBRAS com eficiência é necessário que se utilize uma abordagem baseada nas estruturas mais primitivas da língua. Sendo assim a identificação da configuração de mão é uma parte de suma importância no processo.

3.2 Resumo dos Trabalhos

Esta seção apresenta um resumo dos trabalhos sob a forma de tabela (Tabela 3.1). Alguns autores não divulgam determinados dados, estes são representados na tabela pela sigla N/C (não consta). São apresentados os detalhes relevantes de cada trabalho, como o classificador utilizado, as características e a taxa de acerto reportada pelo autor. Alguns trabalhos apresentaram vários testes e taxas de acerto, nestes casos é considerada a melhor delas.

Tabela 3.1: Tabela Comparativa dos Aspectos mais Relevantes de cada Trabalho.

Trabalho	Abordagem	Tamanho da Base	Classificador	Características	Taxa de Acerto
Marcotti [30]	visualização	21	Árvore de Decisão	Momento de inércia Posição e área da mão	N/C
Wang [52]	visualização	100	HMM	Matriz fundamental da visão estéreo	94%
Starner [45]	visualização	40	HMM	Posição, translação e área da mão	97%
Quan [42]	visualização	5850	SVM	histograma de cores 7 momentos de Hu 128 descritores de Fourier 48 filtros de Gabor	95.55%
Li [29]	visualização	N/C	Template Matching	Esqueleto 3D do Kinect	N/C
Keskin [26]	visualização	N/C	Rede Neural e SVM	Esqueleto 3D e 21 Partes da Mão	99.9%
Mehdi [34]	sensores	24	Rede Neural	Rotação e inclinação da mão Informações dos 5 dedos	88%
Wang [51]	sensores	26	HMM	18 sensores	97%
Zafrulla [54]	sensores	1204	HMM	Informações da mão e acelerômetros	83%
Zafrulla [53]	visualização	N/C	HMM	Esqueleto 3D do Kinect	76.12%
Wang [50]	sensores	5119	HMM	Posição e orientação das mãos Contorno das mãos	100%
Vogler [49]	sensores	53	HMM	Posição 3D dos braços	100%

Os campos N/C (Não Consta) representam informações não divulgadas pelo autor.

3.3 Bases de Dados

Os principais trabalhos encontrados na literatura utilizam bases de dados próprias, ou seja, capturadas pelo autor. Grande parte destas são formadas por vocabulários pequenos e baseados em palavras ou frases capturadas pelo próprio autor e, em geral, não disponíveis para reprodução ou melhoria dos experimentos. Entretanto existem bases de dados públicas destinadas ao reconhecimento de línguas de sinais.

Martinez [32] publicou em 2002 uma extensa base de dados da língua de sinais americana. A base é composta por 2576 vídeos produzidos por 14 usuários da língua de sinais e foi destinada ao treinamento e testes de algoritmos de reconhecimento de padrões. Os vídeos foram capturados na resolução 640x480 pixels, em condições controladas de iluminação. Os sinais que compõe o vocabulário da base foram divididos em duas partes, a primeira delas composta por primitivas da língua e configurações de mão, a segunda parte é formada por vídeos de sentenças da língua americana.

Dreuw [13] fez uma revisão geral das principais bases de dados para rastreamento de mão e face para o reconhecimento de línguas de sinais. A tabela 3.2 apresenta um comparativo entre as bases citadas no trabalho destacando a língua de sinais utilizada, a quantidade de usuários e quantidade de vídeos de cada uma.

Tabela 3.2: Bases de dados citadas por Dreuw [13].

Base de dados	Língua	Usuários	Quantidade de vídeos
Corpus-NGT	Holandesa	92	280
RWTH-BOSTON-50	Americana	3	50
RWTH-BOSTON-104	Americana	3	104
RWTH-BOSTON-400	Americana	5	400
RWTH-PHOENIX-v1.0	Alemã	92	1700
ATIS-ISL	Irlandesa	Vários	680
SIGNUM	Alemã	25	33000

As bases são compostas por vídeos de sinais ou sentenças de línguas de vários países. As bases contém marcações de rastreamento de mão e face quadro a quadro e são aplicáveis no treinamento e desenvolvimento de algoritmos de rastreamento de mão.

Dias et al. [12] publicou um conjunto de dados para treinamento de algoritmos de

reconhecimento da LIBRAS. A base é composta por capturas de 4 atores, cada um executando 3 vezes uma lista de 50 sinais, totalizando 600 amostras. Todos os vídeos foram capturados em formato convencional (RGB), sem o auxílio de luvas ou sensores específicos. O autor extraiu 5 características manualmente: posições vertical e horizontal da mão em relação ao corpo do ator, a configuração de mão, a orientação e a direção da mão. Utilizando estas características foi executado o classificador HMM para comprovar a funcionalidade da base, não são divulgadas as taxas de acerto do sistema.

Todas as bases encontradas contemplam apenas vocabulários isolados, nenhuma delas apresentou recursos para o desenvolvimento de trabalhos utilizando malhas 3D, além disso, nenhuma delas corresponde às configurações de mão da LIBRAS.

CAPÍTULO 4

CLASSIFICAÇÃO DE MALHAS 3D DAS CONFIGURAÇÕES DE MÃO DA LIBRAS

Este capítulo descreve o desenvolvimento do método para classificação das malhas 3D referentes às configurações de mão da LIBRAS. A sequência de execução das tarefas é apresentada na Figura 4.1. Posteriormente, nas seções 4.1 a 4.4, são descritos os detalhes de cada etapa.

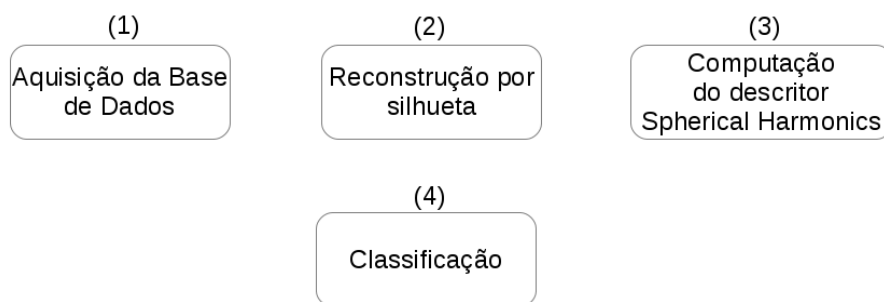


Figura 4.1: Etapas de desenvolvimento do trabalho.

4.1 Aquisição da Base de Dados

Esta seção detalha a criação da base de dados utilizada no desenvolvimento do trabalho. O procedimento utilizado consiste na captura de vídeos das configurações de mão, seleção das visões frontal e lateral, correção de ruídos e geração da malha 3D com o método da silhueta.

O dispositivo de captura utilizado na aquisição da base de dados foi o sensor Kinect, um dispositivo que possibilita a captura de imagens com informações de profundidade que proporciona a captura de imagens em formato RGBD, *Red-Green-Blue-Depth* [46]. A escolha deste sensor se fez devido à possibilidade de aproveitamento de informações do canal de profundidade para uma fácil segmentação da mão.

Inicialmente, para a realização de experimentos com as configurações de mão da

LIBRAS, fez-se necessário o desenvolvimento de softwares para auxiliar a captura dos dados com o sensor Kinect. Estes softwares são descritos a seguir.

Todos os softwares desenvolvidos neste trabalho estão disponíveis para download em <http://web.inf.ufpr.br/vri/alumni/andres-jesse-porfirio-msc-2013/>.

1. NiCapture 1.3:

- Software desenvolvido com o objetivo de auxiliar as capturas das bases de dados;
- Sincroniza a coleta de dados, nomeia corretamente os arquivos de saída e exibe informações aos atores durante a captura.
- O software é uma extensão da ferramenta de captura fornecida pela OpenNI. Os seguintes recursos foram implementados e acrescentados na ferramenta original:
 - Suporte à utilização de um monitor auxiliar para exibição de figuras com exemplos da configuração de mão a ser reproduzida pelo ator;
 - Controles de captura: iniciar e parar captura de um vídeo, avançar e retroceder a configuração de mão;
 - Controle de usuário e tomada: permite a inserção de um prefixo para identificação do usuário e da tomada, utilizados na geração dos arquivos com a nomenclatura CX_Y_Z (detalhada a seguir);
 - Armazenamento do esqueleto 3D para cada quadro do vídeo;
- O padrão de saída dos arquivos é organizado da seguinte maneira:
 - CX_Y_Z, todos os arquivos iniciam com o caractere “C”, indicando se tratar de uma captura de vídeo, em seguida, o campo “X” corresponde ao identificador do ator, o campo “Y” representa a tomada de vídeo e, o campo “Z” representa o identificador da configuração de mão (entre 0 e 60). Um exemplo de arquivo de saída é C4_2_12, trata-se de uma captura do ator “4”, tomada “2”, configuração de mão “12”;
 - Cada captura resulta em dois arquivos com o mesmo nome, diferenciados pela extensão. O primeiro arquivo armazena a captura de vídeo no formato

da OpenNI (".oni"), o segundo arquivo armazena o esqueleto 3D do usuário em um arquivo de texto nomeado ".skl". O esqueleto 3D é calculado pelo Kinect em tempo real e requer calibração prévia do sensor, a fim de evitar a etapa de calibração após a captura o software NiCapture armazena os dados do esqueleto juntamente com o vídeo;

2. NiViewSKL 1.3.1:

- Devido às alterações realizadas no software de captura para o armazenamento do esqueleto 3D juntamente com o vídeo, o software de reprodução padrão da OpenNI tornou-se incompatível com a base de dados, para suprir esta deficiência o mesmo foi alterado de modo ser capaz de trabalhar com os arquivos ".skl";
- O software de reprodução padrão da OpenNI reproduz apenas vídeos no formato ".oni" e realiza a calibração a cada execução. O NiViewSKL eliminou esta calibração, passando a ler os dados do esqueleto do arquivo ".skl";
- Além disto o novo software também realiza a segmentação da mão, fornecendo uma imagem binária contendo apenas a mão do usuário e um fundo preto. Esta funcionalidade visa facilitar a aplicação futura de algoritmos;
- A cada execução o resultado da segmentação da mão é armazenado em sequências de imagens no formato JPEG, cada imagem corresponde a um quadro do vídeo contendo a mão segmentada;
- Em resumo, as alterações implementadas em relação à ferramenta original de reprodução de vídeos da OpenNI são:
 - Capacidade de reproduzir vídeos sem efetuar a calibração;
 - Leitura das informações do esqueleto 3D a partir de um arquivo em formato ".skl";
 - Segmentação da mão;
 - Armazenamento, em formato JPEG, dos quadros do vídeo com a mão segmentada;

Utilizando o software NiCaptura 1.3 foi realizada a aquisição de vídeos das configurações de mão da LIBRAS (base HC-LIBRAS-RGBDS-2011). Posteriormente, para fins de complemento, foi gerada uma base de imagens 2D (HC-LIBRAS-RGB-2011) contemplando apenas a visão frontal da mão.

A primeira base de dados das 61 configurações de mão da LIBRAS foi capturada com o sensor Kinect e é composta por vídeos em formato RGBD acrescidos de arquivos com informações dos esqueletos 3D, as capturas foram realizadas em 2011. A seguir são apresentados detalhes desta base.

1. Materiais:

- Laptop;
- Monitor para a visualização dos sinais;
- Pano azul utilizado como fundo;
- Sensor Kinect;
- Software NiCapture 1.3;

2. Atores:

- 4 usuários sem fluência em LIBRAS;
- 1 usuário conhecedor da LIBRAS supervisionando o treinamento e validando as capturas;

3. Metodologia de Aquisição:

- Montagem do ambiente de captura com o laptop voltado ao usuário responsável pela captura, o sensor e o monitor de visualização voltados ao ator conforme demonstra a Imagem 4.2;
- Treinamento dos usuários:
 - Demonstração das 61 configurações de mãos da LIBRAS (Figura 1.1);
- Requisitos da Captura:

- Instruções de posicionamento à 2m do sensor (Figura 4.3);
 - Instruções de posicionamento mantendo a mão direita no centro do foco da câmera (Figura 4.3);
 - Instruções de sequência de movimentos a serem capturados para cada configuração de mão (Figura 4.4);
 - Instruções de reprodução: Aguardar o final da exibição de um sinal no monitor extra para então reproduzi-lo;
- Etapas da aquisição:
 - (a) Inicialização do software NiCapture 1.3;
 - (b) Atribuição do identificador do usuário (prefixo dos arquivos gerados na captura);
 - (c) Execução do ciclo de gravação: Exibição de uma configuração de mão seguida de sua captura;
 - (d) Repete-se a etapa anterior até que todas as configurações de mão sejam capturadas;

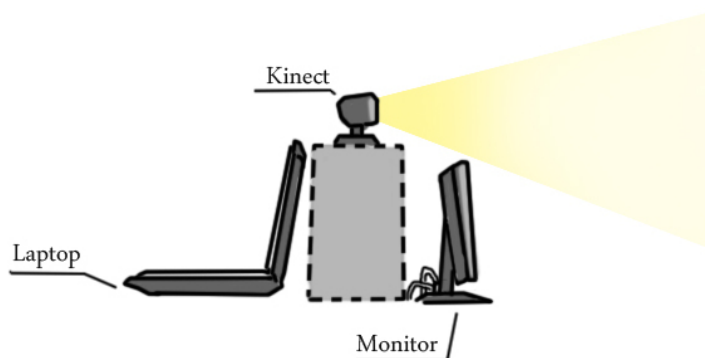


Figura 4.2: Ambiente de captura

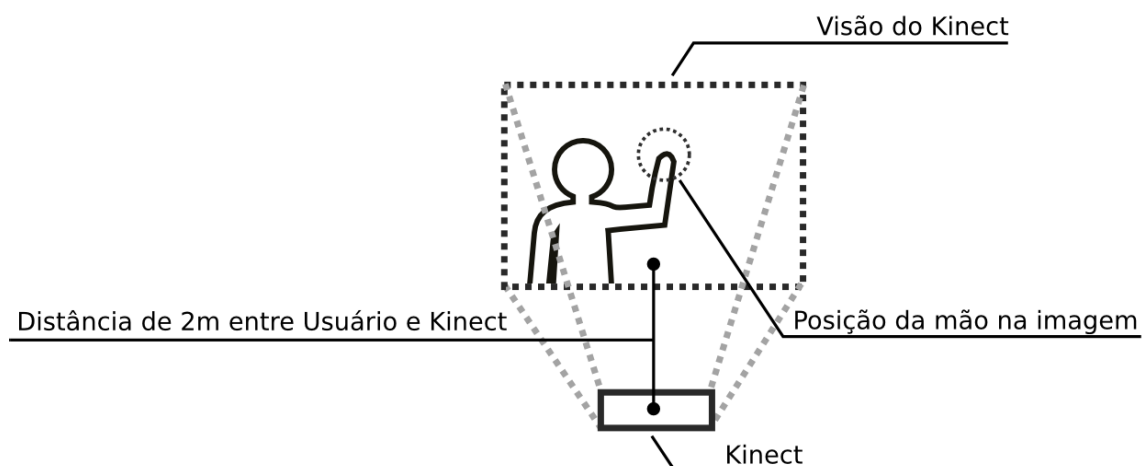


Figura 4.3: Posicionamento do usuário frente ao sensor.

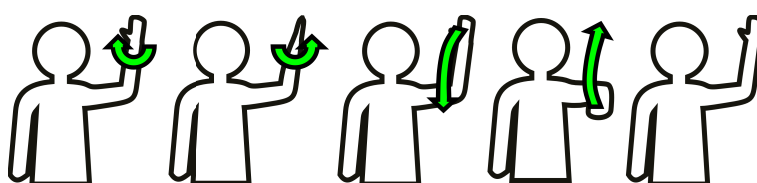


Figura 4.4: Sequencia de movimentos realizada para cada configuração de mão durante a captura.

A base de dados HC-LIBRAS-RGB-2011 foi gerada a partir da base HC-LIBRAS-RGBDS-2011 e é constituída de imagens 2D em formato ".bmp" das 61 configurações de mão da LIBRAS. De cada vídeo da base anterior foi extraído um quadro contendo a mão em visão frontal.

Esta base foi construída como complemento da base HC-LIBRAS-RGBDS-2011, possibilitando também a classificação de configurações de mão da LIBRAS com imagens 2D. A Figura 4.5 apresenta exemplos de imagens da base HC-LIBRAS-RGB-2011;



Figura 4.5: Exemplos de imagens da base de dados HC-LIBRAS-RGBDS-2011.

1. Materiais:

- Base de dados HC-LIBRAS-RGBDS-2011;

- Software de reprodução de vídeos da OpenNI;
- Software de edição de imagens Gimp¹;

2. Metodologia de Construção da Base:

- Reprodução dos vídeos da base HC-LIBRAS-RGBDS-2011 no software da OpenNI;
- Seleção manual de um quadro contendo a visão frontal da mão no canal RGB;
- Captura do quadro;
- Segmentação manual no software Gimp;
- Armazenamento da imagem seguindo o padrão da base de dados, alterando-se apenas a extensão do arquivo para ".bmp";
- Normalização do tamanho das imagens para 200x200 pixels através de script;

O objetivo inicial do trabalho era utilizar unicamente informações do sensor Kinect para o reconhecimento das configurações de mão, entretanto, o sensor e a metodologia de captura não foram capazes de fornecer detalhes suficientes para a geração da malha 3D. Devido à isto a metodologia foi alterada, passando a fazer uso de imagens tratadas manualmente (assume-se, a partir daqui, que a base de dados é artificial).

A reconstrução 3D por silhueta requer imagens de duas ou mais orientações de um objeto, neste caso as posições frontal e lateral da mão. Dado que o dispositivo e os softwares utilizados não são capazes de identificar automaticamente o quadro do vídeo correspondente a estas posições foi necessária a seleção manual dos mesmos. Este processo é detalhado a seguir.

1. Materiais:

- Base de dados HC-LIBRAS-RGBDS-2011;
- Software NiViewSKL 1.3.1;

¹<http://www.gimp.org/>

2. Metodologia de Seleção de Quadros da Base:

- Reprodução dos vídeos da base HC-LIBRAS-RGBDS-2011 no software NiViewSKL 1.3.1;
- Em cada execução o software gera um novo diretório com o video convertido em formato jpeg, uma imagem para cada quadro;
- Todas as imagens são armazenadas com a segmentação já realizada;
- Seleção manual de duas imagens por vídeo: visão frontal e lateral da mão (Figura 4.6);
- Exclusão das imagens que não pertencem à subsequência;
- Ao final do processo devem estar selecionadas 1220 imagens: 610 visões frontais e 610 visões laterais distribuídas uniformemente (10 pares para cada configuração de mão).

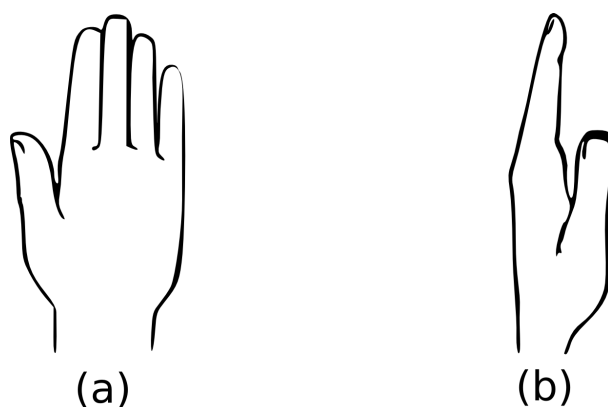


Figura 4.6: Modelo de seleção de quadros, visão frontal (a) e visão lateral (b).

É importante destacar que a inclinação da mão nas duas visões deve ser a mesma, caso contrário o processo de reconstrução por silhueta pode gerar resultados imprecisos. Este problema pode ser minimizado com uma escolha cuidadosa dos quadros a serem segmentados. Em um processo automático seria necessário garantir a mesma inclinação da mão em todas as visões.

Após a seleção dos quadros de interesse, a base de dados foi submetida à uma análise detalhada a fim de localizar imagens com imperfeições ou ruídos que poderiam vir a

comprometer a geração das malhas 3D. Este processo foi realizado manualmente e todas as imagens diagnosticadas impróprias para a reconstrução 3D foram editadas com o objetivo de garantir a execução correta do método de silhueta.

1. Materiais:

- Quadros selecionados da base HC-LIBRAS-RGBDS-2011;
- Software de edição de imagens Gimp²;

2. Metodologia de Correção de Ruídos:

- Análise manual das imagens e diagnóstico de imperfeições;
- Correção das imperfeições no Software Gimp;

As principais causas de defeito nas imagens foram ruídos de segmentação, rotação incorreta e artefatos como a manga da blusa aparecendo na área de interesse. As Figuras 4.7, 4.9 e 4.8 apresentam exemplos de imperfeições (a) e as respectivas correções realizadas (b).

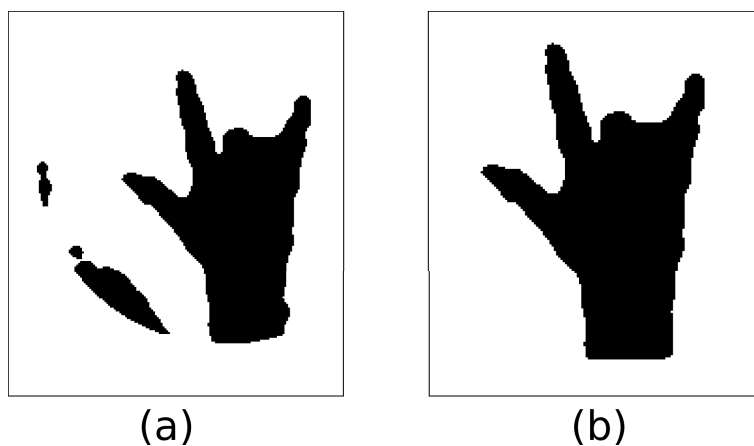


Figura 4.7: Ruído de segmentação (a), imagem corrigida (b).

²<http://www.gimp.org/>

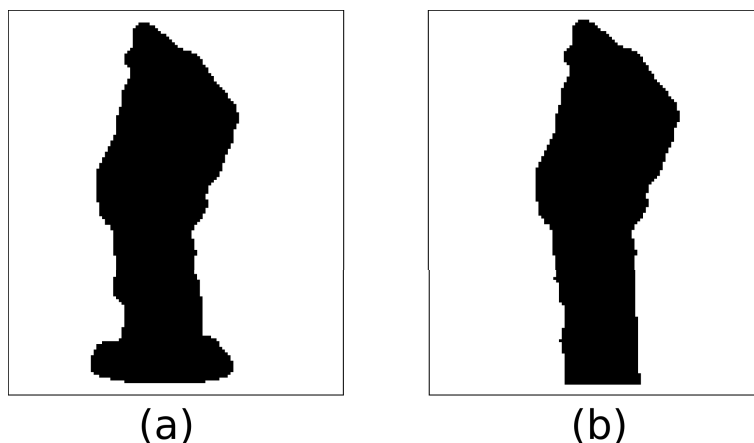


Figura 4.8: Ruído causado pela manga da blusa (a), imagem corrigida (b).

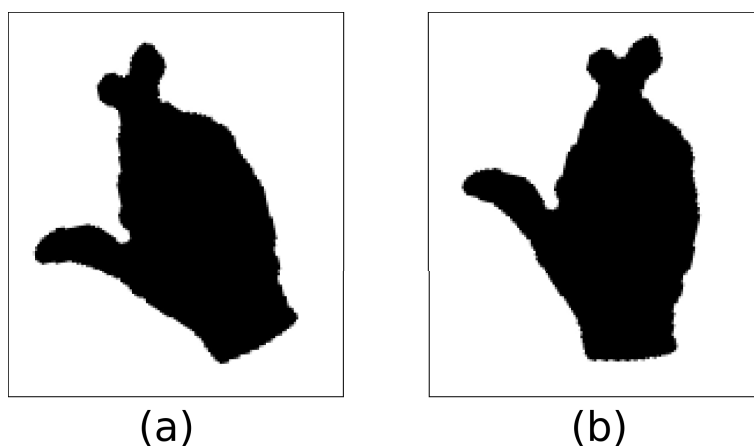


Figura 4.9: Rotação incorreta da mão (a), imagem corrigida (b).

Como é possível observar, os ruídos se apresentam de diversas formas e, nas condições deste trabalho, não existe uma ferramenta capaz de identificá-los automaticamente (ex: um filtro genérico que possa ser aplicado em todas as imagens fornecendo a correção automática dos ruídos). Devido a este fator, a correção de ruídos foi realizada de forma manual.

A correção de ruídos foi realizada com o uso de ferramentas do editor de imagens Gimp. Ruídos de segmentação e artefatos como a manga da blusa (Figuras 4.7 e 4.8) foram removidos com a ferramenta borracha. Imagens com a rotação incorreta foram corrigidos com a ferramenta de rotação, neste caso girando a imagem até atingir a posição correta (Figura 4.9).

De modo geral esta etapa consistiu da edição das imagens para que se tenha uma visão

clara da mão nas posições frontal e lateral, sem ruídos e o mais semelhante possível à uma mão real.

A base de dados utilizada neste trabalho então passa a ser o conjunto de pares de imagens 2D segmentadas e sem ruídos das visões frontal e lateral das 61 configurações de mão da LIBRAS.

4.2 Reconstrução por Silhueta

A reconstrução por silhueta foi implementada sob a forma de script do software Blender 3D, onde dada uma entrada contendo os quadros das visões frontal e lateral da mão previamente selecionadas é gerada uma malha 3D através do método da silhueta.

A fim de garantir um alinhamento mínimo das silhuetas durante a geração da malha, as imagens de entrada foram submetidas à um processo de corte, eliminando áreas sem conteúdo nas bordas superior e inferior da imagem, conforme mostra a Figura 4.10. Este processo tem como objetivo evitar problemas de alinhamento no processo de diferença entre malhas. A Figura 4.11 apresenta duas imagens onde o alinhamento vertical não foi tratado (a), a geração de malha por silhueta com estas imagens (b) e a malha final (c). Nota-se que a operação diferença entre malhas realizou cortes em regiões incorretas.

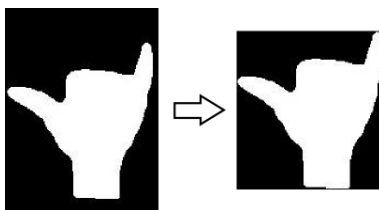


Figura 4.10: Processo de corte para alinhamento das imagens.

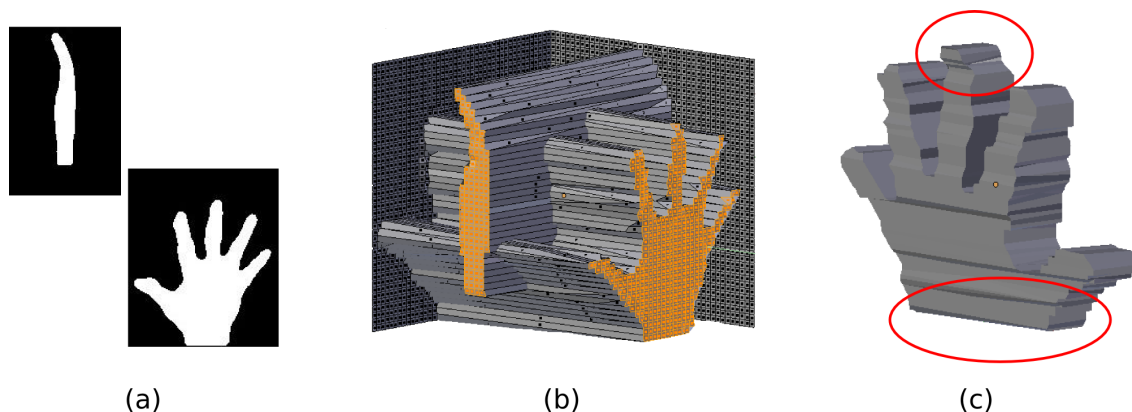


Figura 4.11: Problema ocasionado por alinhamento vertical incorreto das silhuetas: imagens com alinhamento vertical incorreto (a), geração da malha 3D (b) e resultado incorreto da diferença entre malhas (c).

Como foram utilizadas apenas duas imagens na reconstrução por silhueta o deslocamento horizontal, representado por setas na Figura 4.12, não influencia na geração da malha. Este detalhe deve ser tratado quando a reconstrução é baseada em três ou mais imagens.

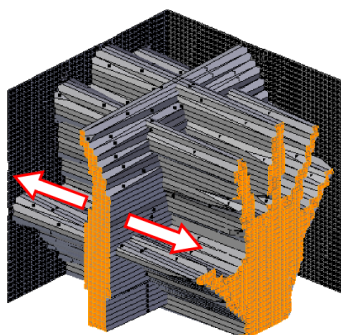


Figura 4.12: Deslocamento horizontal das silhuetas, indiferente quando a técnica utiliza apenas duas visões do objeto.

O software Blender 3D foi escolhido para esta etapa devido ao fato de ser uma ferramenta *OpenSource*³, fornecer recursos para o tratamento e realização de operações como a diferença e pré-processamento de malhas (operações detalhadas no capítulo 2) e, além disso, possibilitar a execução de tarefas em lote (*batch*).

1. Materiais:

- Quadros selecionados e sem ruídos da base HC-LIBRAS-RGBDS-2011;

³Licença do software disponível em <http://www.blender.org/BL/>

- Software Blender 3D 2.6.3a;
- Script de reconstrução por silhueta `lb_reconstruct_blender.py`⁴;

2. Metodologia de Reconstrução 3D:

- Aplicação do script de reconstrução por silhueta nos quadros selecionados e corrigidos;
- Aplicação do filtro de suavização (detalhado na seção 2.1.2);
- Armazenamento da malha convertida no formato nativo do Blender 3D (.blend);
- Armazenamento da malha convertida no formato Stanford (.ply)⁵, formato da extração da característica Spherical Harmonics (seção 4.3);

Conforme citado anteriormente, o algoritmo de reorganização poligonal e suavização aplicado nas malhas é baseado na técnica *Dual Contouring*. Um segundo algoritmo, o *Decimate*, também foi testado para a realização desta tarefa. Este algoritmo tem objetivo similar ao *Dual Contouring*, a reconstrução do mesmo objeto 3D com uma geometria diferente, o *Decimate* entretanto tem como foco principal a redução poligonal [43]. Foi constatado que a utilização deste algoritmo proporciona uma leve suavização das bordas do objeto (Figura 4.13b) em relação ao objeto original (Figura 4.13a) porém mantém o aspecto irregular nos cantos como mostra a Figura 4.13c. Estas características não são interessantes dado que o objetivo desta etapa envolve a transformação da malha 3D em um objeto semelhante à mão real, por este motivo a utilização do algoritmo *Decimate* foi desconsiderada, optando-se unicamente pela utilização do *Dual Contouring*.

⁴Implementado de acordo com os detalhes apresentados na seção 2.1.1.

⁵*Polygon File Format*[33].

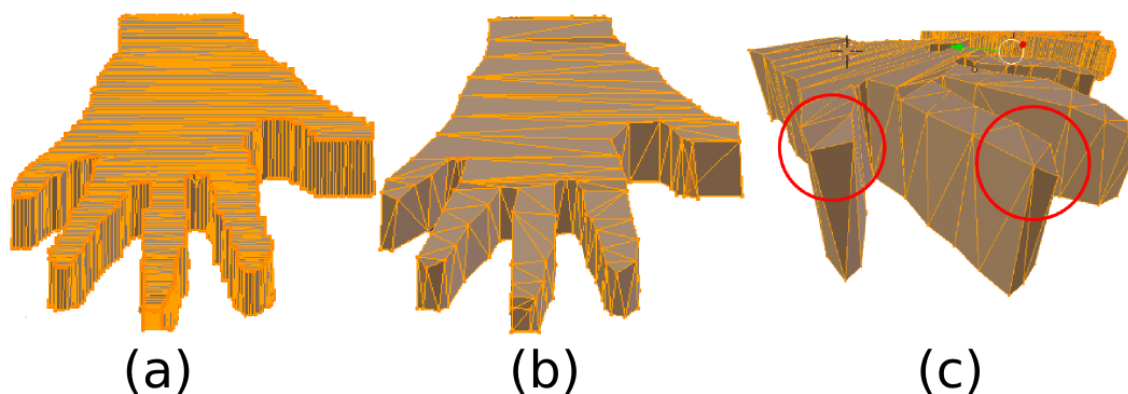


Figura 4.13: Objeto 3D original (a), objeto após a aplicação do algoritmo *Decimate* (b) e cantos irregulares (c).

4.3 Computação do Descritor Spherical Harmonics

A computação do descritor Spherical Harmonics é a etapa de onde a assinatura da malha 3D é extraída para que então seja aplicado o classificador.

1. Materiais:

- Malhas 3D em formato Stanford (.ply);
- Ferramenta de computação do descritor Spherical Harmonics⁶;

2. Metodologia de Computação do Descritor:

- Aplicação da ferramenta de computação do descritor Spherical Harmonics em todos os arquivos .ply;
- Armazenamento da assinatura em formato binário (.sig);
- A assinatura gerada pela ferramenta de Kazhdan é composta por uma matriz de números em ponto flutuante com dimensões 17x32;

4.4 Classificação 3D

Uma vez extraídas as características das malhas 3D é possível realizar o teste de classificação, tendo como objetivo reconhecer configurações de mãos a partir de malhas 3D. O

⁶Ferramenta implementada por Michael Kazhdan, disponível em <http://www.cs.jhu.edu/~misha/HarmonicSignatures/>

classificador foi utilizado com auxílio na biblioteca LibSVM[4].

A biblioteca LibSVM trabalha com um formato de arquivo em texto plano, onde cada linha corresponde à uma instância da base de dados. Inicialmente as características das malhas foram armazenadas em formato binário (.sig), um script foi implementado para convertê-las no formato de entrada da biblioteca LibSVM. O script efetua a leitura dos dados no arquivo .sig e escreve-os em outro arquivo compatível com a biblioteca LIBSVM de acordo com o seguinte procedimento:

1. Para cada Malha 3D \mathbf{Y} contida no conjunto das 610 malhas, geradas com o processo detalhado nas subseções 4.1 e 4.2, é calculada uma matriz $\mathbf{A} \subset \mathbb{R}^2$, de dimensão 17×32 , contendo as características da malha, formadas por 32 assinaturas $SH(f^r)$ (Equação 2.3) e 32 parâmetros de alinhamento.
2. Cada matriz \mathbf{A} , referente à uma malha 3D \mathbf{Y} , é utilizada para gerar a linha z do arquivo em formato LibSVM L^{svm} , tal que:

$$L_z^{svm} = \mathbf{c}_z \ l_0 : f_0, l_1 : f_1, \dots, l_{543} : f_{543} \quad (4.1)$$

na qual L_z^{svm} é a linha de índice z do arquivo L^{svm} , com $z = \{1, 2, \dots, 610\}$, \mathbf{c}_z é a classe da amostra da linha z , l_i e f_i são, respectivamente, o índice e o valor da característica l , sendo $l_i = \{0, 1, \dots, 543\}$ e f_i obtido pela Equação 4.2:

$$f_i = \mathbf{A}_{xy} \left| \begin{array}{l} x = i \bmod 32 + 1 \\ y = i \div 32 + 1 \end{array} \right. \quad (4.2)$$

A utilização do classificador foi baseada em instruções fornecidas pelos próprios desenvolvedores da ferramenta e no guia de Hsu [20], onde são destacadas as melhores práticas para cada tipo de conjunto de dados.

1. Materiais:

- Assinaturas das malhas (.sig)⁷;

⁷Extraídas da base HC_LIBRAS_RGBDS_2011.

- Script de classificação com LibSVM: `lb_test_svm.py`;
- LibSVM versão 3.14;

2. Metodologia de Classificação⁸:

- Conversão das assinaturas em formato `.sig` para o formato do LibSVM (L^{svm});
- Geração de 10 conjuntos de dados: Cada conjunto consiste de uma seleção aleatória de 70% das instâncias para treino e 30% para teste;
- Aplicação do classificador com Ranking 1, 3 e 5 para cada conjunto de dados, totalizando 30 execuções para cada Kernel: RBF e Linear;
- A execução do SVM é realizada com os parâmetros padrão através do script “easy.py” da LibSVM.
- Geração de arquivos de log: resultado de cada instância do conjunto de testes, taxas de acerto e matriz de confusão;

Realizadas as etapas de treinamento e classificação, para cada conjunto de testes, a LibSVM gera um arquivo de saída composto por uma matriz \mathbf{M}^{prob} com os valores das probabilidades para cada amostra. Esta matriz é expressa na Equação 4.3.

$$\mathbf{M}^{prob} = \begin{pmatrix} labels & Lc_1 & Lc_2 & \dots & Lc_n \\ c_1 & p_{1,1} & p_{1,2} & \dots & p_{1,n} \\ c_2 & p_{2,1} & p_{2,2} & \dots & p_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_m & p_{m,1} & p_{m,2} & \dots & p_{m,n} \end{pmatrix} \quad (4.3)$$

Onde: A primeira linha de \mathbf{M}^{prob} é um cabeçalho, formado pela palavra “labels”, seguido das classes Lc_i correspondentes à cada coluna da matriz de probabilidades. Os *labels* são utilizados para a identificação da classe correspondente à cada coluna da matriz uma vez que sua disposição não se dá de forma ordenada. As demais linhas referem-se às probabilidades para cada amostra, sendo: c_i a classe correta da amostra i , m o número

⁸Todos os passos desta etapa são realizados automaticamente pelo script de classificação.

de amostras no conjunto de testes, n o número de classes do problema, p_{ij} o valor da probabilidade da classe com *label* Lc_j para a amostra i .

A classe vencedora CV_i é dada por:

$$CV_i = Lc_{Pmax^i} \quad (4.4)$$

$$Pmax^i = \max_{j=1}^{61}(p_{ij}), \forall i \in \{1, 2, \dots, n\} \quad (4.5)$$

O cálculo CV_i retorna apenas a classe mais provável, ou seja, uma classificação com Ranking 1. O cálculo dos Rankings 3 e 5 é feito da seguinte forma: Uma amostra i é considerada classificada corretamente em Ranking x caso a classe correta de i esteja contida em um subconjunto formado pelas classes correspondentes aos x maiores valores de probabilidades para a amostra i .

A taxa de classificação (TC) de cada teste é calculada pela Equação 4.6:

$$TC = \frac{\sum Ac}{l_{CT}} * 100, \forall Ac \in \mathbf{CT} \quad (4.6)$$

em que Ac são as amostras classificadas corretamente de acordo com o critério apresentado anteriormente, \mathbf{CT} é o conjunto de testes, de tamanho l_{CT} .

A análise da classificação é baseada em 10 execuções provenientes dos 10 subconjuntos de dados. A fim de facilitar a observação do comportamento do classificador é adotada uma estratégia que une os resultados de todos as classificações em um único arquivo, aqui chamado log de classificação.

O log de classificação é composto por uma tabela em formato CSV contendo os dados da classificação e um arquivo de texto contendo a adição das matrizes de confusão de todos os testes. As principais colunas da tabela CSV são: TEST_ID (identificador do subconjunto de dados, variando de 1 a 10), RANKING (variando ente 1, 3 e 5) e HIT_RATE (taxa de acerto do classificador em porcentagem, dado pela Equação 4.6).

O processo de classificação gera 10 matrizes de confusão, as quais são somadas para melhor visualização dos resultados. O cálculo da taxa de reconhecimento do classificador

é a média aritmética das taxas de classificação das 10 execuções para cada Ranking (1, 3 e 5). Assim, com base na tabela CSV do log de classificação, é gerada uma tabela com as médias de acerto para os Rankings 1, 3 e 5 e seus respectivos valores de desvio padrão.

4.5 Classificação 2D

As imagens utilizadas no processo de reconstrução por silhueta (segmentadas e sem ruídos) são utilizadas como entrada para a classificação baseada em características 2D. O objetivo desta etapa é verificar o poder de reconhecimento das configurações de mão da LIBRAS a partir de imagens 2D.

O propósito destes testes é verificar as taxas de reconhecimento das configurações de mão a partir de imagens 2D (como faz a maioria dos autores) na base de dados formada por 1220 imagens segmentadas e sem ruídos (as mesmas utilizadas no processo de geração de malhas pelo método da silhueta).

São avaliados quatro conjuntos de características: os 7 momentos invariantes de Hu [21], as 8 direções de Freeman [17] e as projeções horizontal e vertical de cada imagem [11]. Para fins de normalização as imagens foram redimensionadas para 50x50 pixels antes do cálculo das projeções.

Em 1962 Hu [21] introduziu o uso de momentos invariantes como características para o reconhecimento de padrões visuais. Através do uso de métricas estatísticas Hu definiu características capazes de discriminar padrões geométricos de maneira invariante à posição, orientação e escala. O cálculo desse descritor pode ser efetuado com o uso da função *getHuMoments*, da biblioteca OpenCV⁹. A aplicação da função retorna um vetor de sete posições, que representa a assinatura da imagem.

Herbert Freeman [17] propôs em 1974 uma técnica de descrição de imagens baseada em contornos. A técnica consiste em, dado um pixel inicial do contorno, percorrer todo o objeto rotulando os pixels pertencentes a ele. Por fim, um vetor de oito posições representa a assinatura da imagem, sendo cada índice do vetor correspondente à somatória de todos os

⁹Biblioteca livre para processamento gráfico e visão computacional, disponível em <http://opencv.willowgarage.com>

pixels rotulados com uma determinada direção. A Figura 4.14(a) representa as direções de Freeman e a Figura 4.14(b) apresenta um contorno rotulado de acordo com esta técnica.

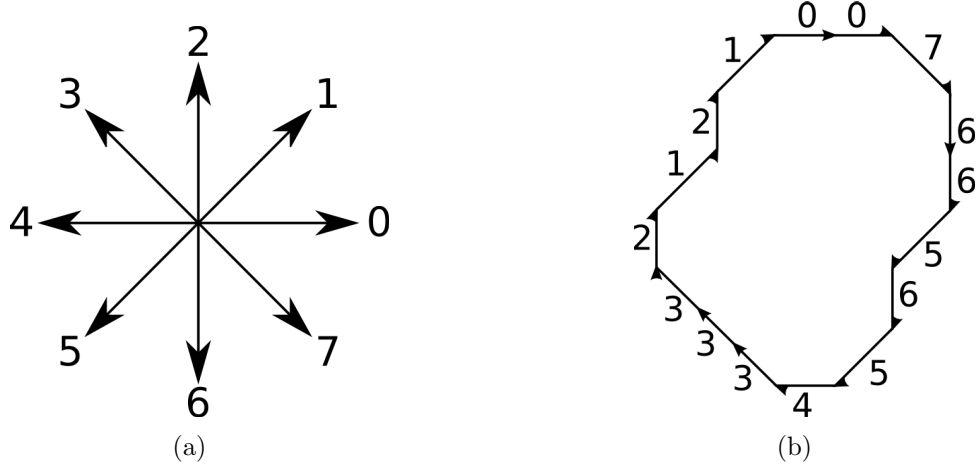


Figura 4.14: Direções de Freeman (a) e Exemplo de contorno rotulado (b).

As projeções de histograma foram introduzidas em 1956 por Glaubergerman apud Trier [47] e são calculadas da seguinte maneira:

Dada uma imagem B , de dimensões $x \times y$, é gerado um vetor $ProjH$, de tamanho x , onde cada termo é dado por:

$$ProjH_i = \sum_{p=1}^x pp \quad (4.7)$$

em que i é cada linha da imagem B e pp é o valor de cada pixel contido na linha i . O vetor $ProjH$ representa a projeção horizontal da imagem B .

A projeção vertical da imagem B é dada por:

$$ProjV_i = \sum_{p=1}^y pp \quad (4.8)$$

em que i é cada coluna da imagem B e pp é o valor de cada pixel contido na coluna i . O vetor $ProjV$, de tamanho y , representa a projeção vertical da imagem B .

A Figura 4.15 apresenta projeções criadas a partir de uma imagem do caractere "5". Os vetores de características gerados por esta técnica não são invariantes à escala, rotação e posição [47].

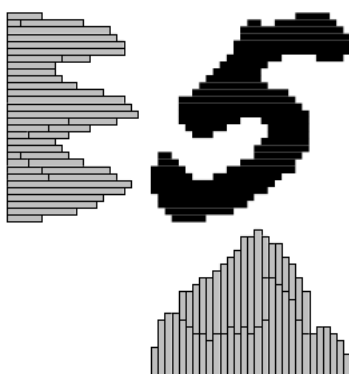


Figura 4.15: Projeções de histograma Horizontal e Vertical[47].

CAPÍTULO 5

RESULTADOS

Os resultados atingidos com este trabalho são apresentados neste capítulo. A seguir são apresentados os resultados referentes à cada uma das etapas citadas no capítulo 4. Por fim, a seção 5.5 apresenta uma comparação de resultados com o método mais similar encontrado na literatura considerações gerais sobre o trabalho.

5.1 Aquisição da Base de Dados

A base de dados foi capturada com êxito, fornecendo um conjunto de vídeos do Kinect das 61 configurações de mão da LIBRAS em várias orientações. Foram necessárias duas tentativas de captura para atingir o resultado esperado, a primeira captura foi realizada com o sensor localizado à uma distância muito grande do ator, com isso a resolução do sensor não foi suficiente para distinguir a configuração de mão em cada vídeo.

Em seguida, com o protocolo de captura atualizado, a base de dados foi capturada novamente, gerando resultados condizentes com a proposta de identificar a configuração de mão. Além disso, na segunda captura foram armazenados os esqueletos 3D juntamente com o vídeo, este processo fornece acesso imediato às junções do corpo em qualquer quadro do vídeo, sem a necessidade de calibração ou rastreamento.

Com relação às bases de dados encontradas na literatura foi observado que nenhuma delas atendia às configurações de mão da LIBRAS com malhas 3D, sendo assim este trabalho acrescenta recursos (como a própria base de dados, os softwares e protocolo de captura) para que novos trabalhos possam ser desenvolvidos.

As principais dificuldades encontradas na captura da base de dados foram: especificação do protocolo de captura, disponibilidade de usuários para as gravações e, principalmente, organização das capturas de modo garantir que todos os usuários executem todas as configurações de mão de modo correto, pois existem muitas poses que se diferem

apenas em pequenos detalhes, podendo ser confundidas facilmente.

Os principais problemas a serem melhorados na base de dados são: rotação incorreta da mão do usuário em algumas capturas (problema da rotação apresentado na Figura 4.9), aparecimento de artefatos como a manga da blusa e, principalmente, a baixa resolução do sensor Kinect (solução apenas com o uso de um sensor melhor).

Para cada vídeo da base HC-LIBRAS-RGBDS-2011 foram selecionados os quadros de interesse: visão frontal e lateral da mão. Os quadros selecionados foram convertidos em imagens 2D no formato jpeg armazenados em um diretório separado, favorecendo a organização da base de dados.

Através da seleção de dois quadros por vídeo obteve-se ao final um conjunto de 1220 imagens, sendo 610 correspondentes às visões frontais e 610 correspondentes às visões laterais.

Todas as imagens foram analisadas e preparadas para a execução do método de reconstrução por silhueta. Artefatos como a manga da blusa foram removidos de todas as imagens onde existiam, imagens com rotação incorreta foram corrigidas e demais ruídos que poderiam causar problemas na reconstrução por silhueta foram corrigidos.

Dado que o dispositivo de captura não fornece uma malha 3D completa e as imagens apresentaram ruídos esta etapa foi crucial para a geração de uma boa base de objetos 3D e, conseqüentemente, na classificação.

Os principais erros encontrados nas malhas foram apresentados na Seção 4.1, este processo foi bastante intrusivo, porém necessário para a geração de boas malhas 3D. A maior parte das imagens precisou de tratamento devido aos diversos tipos de ruídos encontrados. Este processo pode ser evitado com a utilização de meios de captura de imagens mais precisos.

Exemplos de imagens corrigidas para cada configuração de mão são apresentados no Apêndice A. São exibidas as visões frontal e lateral da mão, a seleção das imagens foi feita de forma aleatória respeitando-se a seguinte restrição: uma configuração de mão deve sempre conter duas imagens de um mesmo usuário na mesma tomada.

Existem técnicas e dispositivos em desenvolvimento que, em um futuro próximo,

forneçam meios de obtenção de malhas tridimensionais em tempo real que possibilitarão a classificação direta de malhas 3D, sem a necessidade de reconstrução. Como citado no capítulo 2, um sensor bastante promissor é o LeapMotion.

5.2 Reconstrução por Silhueta

Para cada dupla de imagens frontal/lateral foi gerada uma malha 3D, totalizando 610 reconstruções. Todas as malhas foram submetidas ao filtro de suavização, deste modo a qualidade das mesmas foi melhorada.

As malhas suavizadas foram armazenadas em dois formatos: formato nativo do Blender 3D (.blend) e Stanford (.ply), os arquivos .ply são utilizados na etapa seguinte (cálculo do descritor *Spherical Harmonics*), e os arquivos .blend foram armazenados para permitir futuras aplicações de novos filtros de pré-processamento caso necessário.

O método de reconstrução por silhueta se mostrou eficaz na geração das malhas 3D, porém certos detalhes das mãos, como concavidades, não puderam ser reproduzidos. É possível que a ausência destes detalhes interfira negativamente no processo de classificação e, além disso, é importante ressaltar que sua aplicação em tempo real não é possível nas condições testadas neste trabalho (devido ao fato de se precisar de duas visões da mão, obtidas em instantes de tempo diferentes).

Exemplos de malhas geradas pelo método da silhueta para cada configuração de mão são apresentados no Apêndice B. Os exemplos de cada classe foram escolhidos aleatoriamente e renderizados com o software Blender 3D.

5.3 Computação do Descritor Spherical Harmonics

O uso de scripts python e o Software Blender facilitaram a aplicação da ferramenta de extração do Spherical Harmonics, tornando possível a criação de um procedimento automático onde o software é responsável pela geração da malha em formato Stanford (.ply) e, logo em seguida, a ferramenta atua calculando as assinaturas.

Para cada malha em formato Stanford foi gerado um arquivo binário de assinatura

(.sig), ao todo foram gerados 610 arquivos de assinatura, um para cada malha 3D. A Figura 5.1 apresenta um exemplo de matriz transposta¹ das características extraídas de uma malha 3D referente à configuração de mão 7.

0.0003	4E-006	3E-005	0.0005	0.0021	0.0085	0.023	0.0126	0.1838	0.0037	0.0068	0.0136	0.0134	0.0187	0.0072	0.002	-2E-005
0.0002	2E-006	3E-005	0.0004	0.0022	0.009	0.0161	0.0209	0.1787	0.2142	0.0065	0.0077	0.0133	0.019	0.0071	0.0022	8E-005
0.0002	0.003	2E-005	0.0003	0.0012	0.006	0.014	0.0085	0.0151	0.166	0.0049	0.0083	0.0114	0.0103	0.0077	0.0018	0.0003
0.0002	0.0011	2E-005	0.0003	0.001	0.0056	0.0138	0.0122	0.0195	0.1613	0.1849	0.0076	0.0074	0.0102	0.008	0.0018	0.0003
1E-005	0.0008	0.0125	0.0002	0.0009	0.0027	0.0093	0.0173	0.0279	0.0159	0.1264	0.0051	0.0079	0.0081	0.0049	0.0019	0.0003
8E-006	0.0028	0.0048	0.0001	0.0007	0.002	0.0104	0.0137	0.0199	0.0473	0.1152	0.1232	0.0069	0.0063	0.0048	0.002	0.0003
2E-006	0.0007	0.0028	0.0335	0.0009	0.0027	0.0044	0.0084	0.015	0.0586	0.0131	0.0783	0.0046	0.0062	0.0039	0.0014	0.0003
3E-006	0.0002	0.0097	0.0182	0.0006	0.0021	0.0039	0.0115	0.011	0.0362	0.0614	0.0593	0.0628	0.0044	0.0032	0.0013	0.0003
1E-006	9E-005	0.004	0.0114	0.07	0.0018	0.0038	0.0077	0.0065	0.0255	0.0784	0.0114	0.0376	0.0042	0.0031	0.0011	0.0002
2E-006	7E-005	0.0025	0.0203	0.0504	0.0016	0.0039	0.0084	0.0133	0.0122	0.0403	0.0561	0.0202	0.0234	0.002	0.0009	0.0002
7E-007	4E-005	0.0015	0.0094	0.0333	0.117	0.0021	0.0046	0.0083	0.0125	0.0412	0.0714	0.0084	0.0133	0.0023	0.0009	0.0002
1E-006	3E-005	0.0006	0.0091	0.0266	0.0989	0.0027	0.0045	0.0093	0.0145	0.0225	0.0347	0.0358	0.0041	0.006	0.0006	0.0002
5E-007	2E-005	0.0004	0.0059	0.0197	0.0704	0.1581	0.0032	0.0058	0.0063	0.0215	0.0396	0.044	0.0041	0.0033	0.0007	0.0002
9E-007	1E-005	0.0003	0.003	0.0203	0.0199	0.1479	0.0025	0.0054	0.0114	0.0135	0.0288	0.0246	0.0156	0.0004	0.001	0.0001
3E-007	1E-005	0.0003	0.0025	0.0123	0.0319	0.1216	0.1843	0.0053	0.0078	0.0091	0.0283	0.0257	0.0181	0.0012	0.0006	0.0001
6E-007	2E-005	0.0003	0.0011	0.0066	0.0264	0.0135	0.1773	0.0033	0.006	0.0137	0.0134	0.0244	0.0121	0.0045	-2E-005	0.0001
3E-007	1E-005	0.0002	0.0013	0.0062	0.0171	0.0294	0.1724	0.2109	0.0063	0.0083	0.0129	0.0246	0.012	0.0049	0.0002	6E-005
0.0011	1E-005	0.0001	0.0007	0.0038	0.011	0.0158	0.0132	0.179	0.004	0.0076	0.0126	0.0124	0.0128	0.0038	0.0008	-8E-006
0.0005	1E-005	8E-005	0.0006	0.0036	0.0117	0.0138	0.0126	0.1736	0.2053	0.007	0.0074	0.0123	0.013	0.0038	0.0009	2E-005
0.0005	0.0066	7E-005	0.0005	0.002	0.0081	0.0166	0.014	0.0161	0.1479	0.0053	0.0083	0.01	0.0076	0.0041	0.0007	1E-004
0.0011	0.0023	6E-005	0.0005	0.0016	0.0082	0.0144	0.0141	0.0346	0.1413	0.1559	0.0076	0.0073	0.0074	0.0043	0.0008	1E-004
0.0001	0.0014	0.0213	0.0005	0.0017	0.0033	0.0094	0.0162	0.0434	0.0147	0.1025	0.0047	0.0073	0.006	0.0028	0.0008	9E-005
4E-005	0.0056	0.0097	0.0004	0.0013	0.0026	0.0113	0.012	0.0283	0.0565	0.0866	0.0911	0.0058	0.0048	0.0027	0.0009	9E-005
1E-005	0.0019	0.0058	0.0498	0.0013	0.0034	0.0062	0.0064	0.0179	0.0711	0.0122	0.0561	0.0046	0.0047	0.0023	0.0006	1E-004
1E-005	0.0009	0.0149	0.0317	0.0011	0.003	0.0063	0.0122	0.0101	0.0401	0.0613	0.0367	0.04	0.0031	0.0018	0.0006	0.0001
4E-006	0.0005	0.0065	0.0205	0.0931	0.0021	0.0043	0.0086	0.0089	0.0346	0.0785	0.0103	0.0233	0.0033	0.0018	0.0005	8E-005
6E-006	0.0002	0.0051	0.0247	0.0736	0.0023	0.0046	0.0095	0.0139	0.0172	0.0381	0.047	0.0097	0.0125	0.0011	0.0004	7E-005
6E-006	0.0001	0.0033	0.0135	0.0499	0.1391	0.0021	0.0054	0.0074	0.0169	0.0428	0.0589	0.0063	0.007	0.0014	0.0004	7E-005
1E-005	0.0001	0.0016	0.0145	0.0246	0.1246	0.0027	0.0045	0.0098	0.0141	0.0267	0.0303	0.0248	0.0014	0.0026	0.0003	6E-005
7E-006	1E-004	0.0012	0.0089	0.0267	0.0947	0.1733	0.0045	0.0069	0.0064	0.0258	0.0332	0.0297	0.0024	0.0014	0.0003	6E-005
1E-005	7E-005	0.0005	0.0048	0.025	0.0158	0.166	0.0028	0.0058	0.0132	0.0132	0.0281	0.0182	0.0088	5E-005	0.0004	4E-005
5E-006	4E-005	0.0006	0.0041	0.0155	0.0332	0.149	0.1982	0.0057	0.0083	0.0115	0.0281	0.0184	0.0099	0.0006	0.0002	5E-005

Figura 5.1: Matriz transposta das características de uma malha 3D referente à configuração de mão 7.

5.4 Classificação

Foram realizados testes de classificação utilizando o descritor Spherical Harmonics (subseção 5.4.1) e utilizando características 2D (subseção 5.4.2). Além disso, foi executado um teste para avaliar o impacto da suavização das malhas na etapa de classificação (subseção 5.4.3) e testes com números reduzidos de atributos (subseção 5.4.4). Na subseção 5.4.5 encontra-se uma análise das matrizes de confusão do teste principal (descritor *Spherical Harmonics* no problema com 61 classes).

¹A matriz foi transposta apenas para melhor visualização nesta página.

5.4.1 Esféricos Harmônicos em 61 Classes

Dadas as assintauras de malhas em formato .sig, foi gerado um único arquivo em formato LibSVM onde cada linha corresponde à uma instância. O arquivo foi subdivido aleatoriamente 10 vezes, gerando 10 subconjuntos de dados, cada um com um arquivo de treino (70% das instâncias da base) e um de testes (30% das instâncias da base).

Os testes foram executados nos 10 subconjuntos de dados com Ranking 1, 3 e 5 utilizando duas configurações do SVM: Kernel Linear e Kernel RBF. As Tabelas 5.1 e 5.2 apresentam as médias das execuções e o desvio padrão para cada Ranking.

Tabela 5.1: Média de 10 execuções do SVM com Kernel RBF.

Ranking	Taxa de acerto (%)	Desvio Padrão (%)
1	85.68	1.65
3	96.67	1.27
5	98.52	0.86

Tabela 5.2: Média de 10 execuções do SVM com Kernel Linear.

Ranking	Taxa de acerto (%)	Desvio Padrão (%)
1	86.06	2.06
3	96.83	1.31
5	98.57	0.60

5.4.2 Características 2D

Utilizando-se apenas as imagens 2D das visões frontal e lateral, segmentadas e sem ruídos, foram realizados testes de classificação utilizando as técnicas clássicas.

A base de dados, formada por 1220 imagens, sendo 610 visões frontais e 610 laterais, foi subdivida em proporção 70/30 da mesma forma como a base de malhas 3D. A taxa de classificação foi baseada na média aritmética de 10 execuções, os resultados são expressos na Tabela 5.3. Em todos os casos o vetor de características foi formado pela concatenação dos vetores correspondentes às visões frontal e lateral da mão.

Tabela 5.3: Média de 10 execuções do SVM com características 2D.

Ranking	Característica	Taxa de Acerto (%)	Desvio Padrão (%)
1	Projeção Horizontal	83.42	1.09
1	Projeção Vertical	74.78	2.17
1	Momentos de HU	56.73	3.01
1	Direções de Freeman	6.57	4.91
3	Projeção Horizontal	98.36	0.48
3	Projeção Vertical	94.61	0.89
3	Momentos de HU	82.66	3.21
3	Direções de Freeman	13.26	9.99
5	Projeção Horizontal	99.34	0.52
5	Projeção Vertical	97.55	0.88
5	Momentos de HU	90.81	2.53
5	Direções de Freeman	18.26	13.91

Em geral, taxas de reconhecimento foram piores quando utilizando características 2D. Percebe-se que a utilização do descritor Spherical Harmonics em malhas 3D, além de fornecer benefícios quanto à invariância à diversos parâmetros e resolver o problema da oclusão ainda fornece um poder maior de reconhecimento.

Destaca-se ainda que a aquisição de imagens segmentadas e sem ruídos a partir de sequências de vídeos de uma conversação em LIBRAS é sujeita aos problemas clássicos de segmentação encontrados pela maioria dos autores. Além disso, os descritores projeção horizontal e vertical (que obtiveram melhores resultados de reconhecimento) não são invariantes à rotação, assim, surge um novo problema: detectar a orientação correta da mão. Com base nisso, pode-se afirmar que a utilização de descritores 3D como o *Spherical Harmonics* oferece mais vantagens em relação aos descritores 2D avaliados.

5.4.3 Impacto da Suavização das Malhas na Classificação

Foi realizado um teste de classificação para analisar o impacto da suavização e reorganização poligonal das malhas (Subseção 2.1.2 do capítulo 2). O teste foi executado em condições idênticas ao da Tabela 5.1 (Kernel RBF e parâmetros padrão). O descritor *Spherical Harmonics* foi extraído da mesma maneira, porém nenhuma malhas recebeu o filtros de pré-processamento e reorganização poligonal (que também suaviza o objeto 3D). Os resultados são expressos na Tabela 5.4.

Tabela 5.4: Média das execuções do SVM nos 10 Subconjuntos de Dados com Kernel RBF sem Pré-processamento das Malhas.

Ranking	Taxa de acerto (%)	Desvio Padrão (%)
1	84.64	2.29
3	95.46	1.12
5	98.08	0.88

Nota-se que as taxas de acerto do classificador são muito similares (menos de 1% de diferença no Ranking 1, caso onde existe maior diferença) às taxas alcançadas com as malhas pré-processadas, conclui-se que esta etapa não apresentou benefícios para a melhoria da taxa de reconhecimento.

Ressalta-se que o objetivo do pré-processamento não incluía melhoramento da taxa de reconhecimento do classificador, mas sim possibilitar que o objeto de estudo do trabalho (as malhas 3D) fossem parecidas com a mão real, simulando um sistema onde a aquisição da malha 3D é precisa e o objeto virtual é igual (ou muito semelhante) ao objeto original.

5.4.4 Redução de Atributos

Também foram realizadas execuções do classificador com um número reduzido de atributos. A redução de atributos, em alguns casos, pode melhorar o desempenho do classificador, eliminando características pouco ou nada influentes [28]. Foi utilizada a ferramenta de redução de atributos da LibSVM² com os parâmetros padrão, as quantidades de atributos avaliadas foram: 544, 272, 136, 68, 34, 17, 8 e 4.

O algoritmo utilizado para a escolha dos atributos mais relevantes é baseado na estratégia *F-Score* [5]. A quantidade de atributos original era 544, em todos os casos a utilização de menos atributos piorou o resultado, levando à conclusão de que neste caso essa técnica não apresenta benefícios.

²http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/#feature_selection_tool

5.4.5 Análise das matrizes de confusão

O passo seguinte para a detecção dos erros de classificação foi a análise das matrizes de confusão, para melhor visualização dos resultados foi gerada uma única matriz $M^{conf} \subset \mathbb{R}^2$ contendo a adição de todas as 10 matrizes provenientes de cada execução do classificador como mostra a Equação 5.1, onde M_i é a matriz de confusão resultante do teste i com Ranking 1, apenas este valor de Ranking foi considerado nesta etapa. Os valores são relativos à execução do SVM com Kernel RBF.

$$M^{conf} = \sum_{i=1}^{10} M_i \quad (5.1)$$

Utilizando esta estratégia uma única matriz é capaz de representar as 10 execuções, facilitando a análise dos dados. Através da análise de M^{conf} , foi possível observar que existem várias configurações de mão muito parecidas entre si (como os pares 18/19 e 20/21 na Figura 1.1) que em muitos casos são confundidas pelo SVM. A Figura 5.2 apresenta a matriz M^{conf} .

[illegible]

Figura 5.2: Matriz de confusão das execuções do Kernel RBF (M^{conf}) com Ranking 1, onde as linhas e colunas representam as 61 configurações de mão da LIBRAS. Destaque às regiões das configurações de mão 18/19, 20/21, 23, 24 e 28 onde houve grande índice de erro.

As Figuras 5.3 a 5.5 apresentam exemplos de malhas 3D correspondentes às configurações de mão 18 e 19, 20 e 21 e 24 e 28 (exemplos de áreas onde houve grande índice de erro). É possível perceber que a similaridade visual e estrutural (organização dos polígonos) entre elas é grande, o que causa erro no classificador. Em LIBRAS existem várias configurações de mão similares, que durante uma conversa podem até ser confundidas por um ser humano, neste caso o contexto da conversa e os outros parâmetros globais (como o movimento) contribuem para o entendimento do sinal, a mesma estratégia é aplicável à sistemas computacionais (através do cruzamento de resultados de vários classificadores de parâmetros globais).



Figura 5.3: Malhas 3D referentes às configurações de mão 18 e 19.



Figura 5.4: Malhas 3D referentes às configurações de mão 20 e 21.



Figura 5.5: Malhas 3D referentes às configurações de mão 24 e 28.

Além disso, a ausência de detalhes na malha 3D como concavidades causada pela técnica de silhueta omite informações que poderiam distinguir tais configurações de mão. Este problema pode ser corrigido com a utilização de malhas mais precisas provenientes de outro sensor ou técnica, melhorando a taxa de acerto do classificador.

Os principais fatores que influenciaram na geração de malhas 3D similares foram: método de reconstrução 3D incapaz de reproduzir determinados detalhes, erro humano na edição das imagens para correção de ruídos, dispositivo de aquisição com resolução insuficiente e, por fim, as próprias configurações de mão nativamente parecidas.

5.5 Comparação de Resultados e Considerações

Na literatura o trabalho que mais se assemelha à classificação de configurações de mão aqui desenvolvido é o de Keskin [26]. O resultado apresentado pelo autor é muito bom (99.9% de acerto), entretanto deve-se levar em consideração que o mesmo utilizou apenas as configurações de mão referentes aos 10 dígitos, totalizando 10 classes na classificação frente às 61 da LIBRAS utilizadas neste trabalho.

Para fins de comparação de resultados foi realizado um teste semelhante, da base de configurações de mão da LIBRAS foram selecionadas apenas malhas 3D referentes aos 10 dígitos. As configurações de mão (representadas na Figura 1.1) utilizadas para formar os dígitos na LIBRAS, segundo Pimenta e Quadros [16], são expressos na Tabela 5.5. Para

realizar a classificação de 10 configurações de mão como fez Keskin [26] a configuração 25 que se repete para os dígitos 6 e 9 foi utilizada apenas uma vez, a segunda ocorrência foi substituída pela configuração de mão 60, desta forma o classificador trabalha com 10 classes distintas.

Tabela 5.5: Configurações de mão da LIBRAS referentes aos dígitos.

Dígito	0	1	2	3	4	5	6	7	8	9
Configuração de Mão	22	2	38	51	55	31	25	9	7	25

A classificação de 10 configurações de mão da LIBRAS foi realizada com os parâmetros padrão do SVM e kernel RBF, os resultados foram semelhantes aos de Keskin, conforme mostra a tabela 5.6.

Tabela 5.6: Média das execuções do SVM nos 10 Subconjuntos de Dados com 10 classes.

Ranking	Taxa de acerto (%)	Desvio Padrão (%)
1	98.33	1.66
3	100	0
5	100	0

Os demais trabalhos fizeram uso de características diferentes, como o movimento, e focaram em objetivos diferentes, como o reconhecimento de sinais ou sentenças e não configurações de mão, desta forma a comparação de taxas de acerto com estes trabalhos não é coerente.

A utilização do descritor Spherical Harmonics se mostrou eficiente para a classificação das malhas 3D das configurações de mão da LIBRAS e, além disso, pode-se perceber por meio da análise feita por Clark et al. [8] que o método é superior a outros encontrados na literatura, portanto sua aplicação foi satisfatória para a classificação de malhas 3D. Pode-se destacar que, mesmo não atingindo uma classificação perfeita na base de dados utilizada, o reconhecimento da configuração de mão a partir da malha 3D é uma contribuição importante para o desenvolvimento de uma solução mais complexa onde o objetivo é reconhecer qualquer sinal ou sentença da LIBRAS.

Uma vez que se tenha soluções para o reconhecimento de todos os parâmetros globais, mesmo que nenhum alcance 100%, é possível criar uma união de todos e, com resultados

de vários classificadores e informações sobre o contexto da conversa, identificar o sinal correto com precisão.

Além disso, a classificação de malhas 3D mostrou-se promissora e viável e, uma vez que se tenha um sensor adequado ou técnicas capazes de fornecer malhas mais precisas, este sistema pode ser aplicado em ambientes reais, sem preocupações com problemas de luminosidade e oclusão como ocorrem com imagens 2D.

CAPÍTULO 6

CONCLUSÃO

Foi observado que o reconhecimento de sinais têm sido bastante estudado na literatura, porém a maior parte dos trabalhos é de difícil utilização em ambientes reais e, em sua maioria, baseiam-se em abordagens inapropriadas (como a utilização de palavras isoladas).

Com base nos conceitos da abordagem de Antunes et al. [2] este trabalho foi focado no reconhecimento do parâmetro global configuração de mão da LIBRAS. Uma base de dados foi construída para a realização dos experimentos e o uso de malhas 3D e o descritor *Spherical Harmonics* obteve resultado satisfatório.

As principais contribuições do trabalho foram: a criação de uma base de vídeos, em formato RGBDS, das configurações de mão da LIBRAS, acrescida de imagens 2D referentes à cada um dos vídeos onde a mão aparece em visão frontal. Além disso, foi criada uma base de malhas 3D com o método da silhueta, baseado em duas visões da mão, a qual foi submetida à um processo de extração de características e, posteriormente, classificação das 61 configurações de mão.

Foram observados desafios, como a dificuldade na geração das malhas e a deficiência da tecnologia empregada neste trabalho, entretanto, foi demonstrada a viabilidade do reconhecimento das configurações de mão a partir de malhas 3D e apresentados os seus benefícios. Também foi apresentado um comparativo com a classificação utilizando características 2D, onde os resultados foram favoráveis à utilização de malhas 3D.

É previsível que o avanço da tecnologia (novos dispositivos e técnicas) facilitará a captura de cenas em 3D em um futuro próximo, isto tornará possível a captura em tempo real da malha 3D da mão, permitindo o reconhecimento contínuo do parâmetro configuração de mão e, conseqüentemente, contribuindo para o reconhecimento total da LIBRAS por computadores.

De modo geral alterações no modo de aquisição da malha 3D não implicarão em

descarte do método apresentado, melhorias na aquisição fornecerão mais detalhes às malhas e, conseqüentemente, possibilitarão uma classificação mais precisa. Além disso, um sistema que trabalha com malhas 3D de configurações de mão pode ser reutilizado no reconhecimento de línguas de sinais visto que trabalha com um parâmetro primitivo da língua.

Em resumo, o presente trabalho demonstrou a eficácia da classificação de configurações de mão a partir de malhas 3D, fornecendo um recurso poderoso para futuros sistemas de reconhecimento de línguas de sinais que podem seguir esta abordagem e evitar problemas de oclusão, iluminação e ainda obter detalhes de todos os lados da mão em qualquer orientação, detalhes estes que podem fazer toda a diferença no reconhecimento de um sinal.

BIBLIOGRAFIA

- [1] B. Aisen. A comparison of multiclass svm methods, 2006.
- [2] D. R. Antunes, C. Guimaraes, L. S. Garcia, L. E. S. Oliveira, e S. Fernandes. A framework to support development of sign language human-computer interaction: Building tools for effective information access and inclusion of the deaf. *Proc. Fifth Int Research Challenges in Information Science (RCIS) Conf*, páginas 1–12, 2011.
- [3] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
- [4] C. Chang e C. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.
- [5] Y. Chen e C. Lin. Combining svms with various feature selection strategies. *Taiwan University*. Springer-Verlag, 2005.
- [6] G.K.M. Cheung, T. Kanade, J.-Y. Bouguet, e M. Holler. A real time system for robust 3d voxel reconstruction of human motions. *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, páginas 714 –720 vol.2, 2000.
- [7] K. M. Cheung, S. Baker, e T. Kanade. Visual hull alignment and refinement across time: A 3d reconstruction algorithm combining shape-from-silhouette with stereo. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June de 2003.
- [8] D. E. R. Clark, J. R. Corney, F. Mill, H. J. Rea, A. Sherlock, e N. K. Taylor. Benchmarking shape signatures against human perceptions of geometric similarity. *Computer-Aided Design*, 38(9):1038–1051, 2006.
- [9] C. Cortes e V. Vapnik. Support-vector networks, 1995.

- [10] C. Cortes e V. Vapnik. Support-vector networks. *Machine Learning*, páginas 273–297, 1995.
- [11] R. M. O. Cruz, G. D. C. Cavalcanti, e T. I. Ren. "análise de técnicas de extração de características para o reconhecimento de dígitos manuscritos.". *Sibgrapi*, 2009.
- [12] K. P. de; Pistori H. Dias, J. B.; Souza. Conjunto de treinamento para algoritmos de reconhecimento de libras. *II Workshop de Visão Computacional, São Carlos*, 2006.
- [13] P. Dreuw, J. Forster, e H. Ney. Tracking benchmark databases for video-based sign language recognition. *ECCV International Workshop on Sign, Gesture, and Activity*, Crete, Greece, setembro de 2010.
- [14] P. Dreuw e H. Ney. Visual modeling and feature adaptation in sign language recognition. *Voice Communication (SprachKommunikation), 2008 ITG Conference on*, páginas 1–4, oct. de 2008.
- [15] P. Dreuw, D. Stein, T. Deselaers, D. Rybach, M. Zahedi, J. Bungeroth, e H. Ney. Spoken language processing techniques for sign language recognition and translation. *Technology and Dissability*, 20(2):121–133, 2008.
- [16] Nelson Pimenta e Ronice Muller de Quadros. *Curso LIBRAS 1 4a Edição*. Editora Vozes, 2010.
- [17] H. Freeman. Computer processing of line-drawing images. *ACM Comput. Surv.*, 6:57–97, March de 1974.
- [18] C. Guimaraes, D. R. Antunes, D. F. G. Trindade, R. A. L. Silva, e L. S. Garcia. Structure of the brazilian sign language (libras) for computational tools: Citizenship and social inclusion. *Organizational, Business, and Technological Aspects of the Knowledge Society*. Springer Berlin Heidelberg, 2010.
- [19] T. Hastie e R. Tibshirani. Classification by pairwise coupling. *The Annals of Statistics*, vol. 26, páginas 451–471, 1998.

- [20] C. Hsu, C. Chang, e C. Lin. A practical guide to support vector classification, 2010.
- [21] M. Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179–187, february de 1962.
- [22] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, e A. Fitzgibbon. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. *Proceedings of the 24th annual ACM symposium on User interface software and technology, UIST '11*, páginas 559–568, New York, NY, USA, 2011. ACM.
- [23] T. Ju, F. Losasso, S. Schaefer, e J. Warren. Dual contouring of hermite data, 2002.
- [24] M. Kazhdan e T. Funkhouser. Harmonic 3d shape matching. *ACM SIGGRAPH 2002 conference abstracts and applications, SIGGRAPH '02*, páginas 191–191, New York, NY, USA, 2002. ACM.
- [25] M. Kazhdan, T. Funkhouser, e S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, 2003.
- [26] C Keskin, F. Kiraç, Y. E. Kara, e L. Akarun. Real time hand pose estimation using depth sensors. *ICCV Workshops*, páginas 1228–1234. IEEE, 2011.
- [27] H. Kim, S. Yang, e K. Sohn. 3d reconstruction of stereo images for interaction between real and virtual worlds. *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality, ISMAR '03*, páginas 169–, Washington, DC, USA, 2003. IEEE Computer Society.
- [28] D. Lara, A. Porfirio, e A. Pozo. Otimização por colônia de abelhas aplicada ao problema de seleção de atributos. *Anais do III EATI - Encontro Anual de Tecnologia da Informação*, 2012.

- [29] K. F. Li, K. Lothrop, E. Gill, e S. Lau. A web-based sign language translator using 3d video processing. *Network-Based Information Systems (NBIS), 2011 14th International Conference on*, páginas 356–361, sept. de 2011.
- [30] P. Marcotti, L. B. Abiuzi, P. M. S. R. Rizol, e C. H. Q. Forster. Interface para reconhecimento da língua brasileira de sinais. *XVIII Simpósio Brasileiro de Informática na Educação - SBIE - Mackenzie*, 2007.
- [31] F. Mariusz e M. Szymon. On the use of graph parsing for recognition of isolated hand postures of polish sign language. *Pattern Recogn.*, 43:2249–2264, June de 2010.
- [32] A. M. Martinez, R. B. Wilbur, R. Shay, e A. C. Kak. Purdue rvl-slll asl database for automatic recognition of american sign language. *Proc. Fourth IEEE Int Multimodal Interfaces Conf*, páginas 167–172, 2002.
- [33] K. McHenry e P. Bajcsy. An overview of 3d data content, file formats and viewers. 2008.
- [34] S. A. Mehdi e Y. N. Khan. Sign language recognition using sensor gloves. *Proc. 9th Int. Conf. Neural Information Processing ICONIP '02*, volume 5, páginas 2204–2206, 2002.
- [35] D. Meyer. Support vector machines: The interface to libsvm in package e1071. Relatório técnico, Technische Universitat Wien, Austria, 2012.
- [36] J. Milgram, M. Cheriet, e R. Sabourin. "One Against One" or "One Against All": Which One is Better for Handwriting Recognition with SVMs? Guy Lorette, editor, *Tenth International Workshop on Frontiers in Handwriting Recognition*, La Baule (France), outubro de 2006. Université de Rennes 1, Suvisoft. <http://www.suvisoft.com> Université de Rennes 1.
- [37] A. Y. Mülayim, U. Yilmaz, e V. Atalay. Silhouette-based 3d model reconstruction from multiple images, 2003.

- [38] L. R. C. Ormonde. Classificação automática de páginas web multi-label via mdl e support vector machines. Dissertação de Mestrado, Universidade de Brasília, 2009.
- [39] S. M. Peres, F. C. Flores, D. Veronez, e C. J. M. Olguin. Libras signals recognition: a study with learning vector quantization and bit signature. *Proc. Ninth Brazilian Symp. Neural Networks SBRN '06*, páginas 119–124, 2006.
- [40] John C. Platt. *Probabilities for SV Machines*, páginas 61–74. MIT Press, 2000.
- [41] D. Price, S. Knerr, L. Personnaz, e G. Dreyfus. Pairwise neural network classifiers with probabilistic outputs. *Neural Information Processing Systems, MIT Press*,, páginas 1109–1116, 1995.
- [42] Y. Quan. Chinese sign language recognition based on video sequence appearance modeling. *Proc. 5th IEEE Conf. Industrial Electronics and Applications (ICIEA)*, páginas 1537–1542, 2010.
- [43] W. J. Schroeder, J. A. Zarge, e W. E. Lorensen. Decimation of triangle meshes. *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '92, páginas 65–70, New York, NY, USA, 1992. ACM.
- [44] M. Shantz e H. Poizner. A computer program to synthesize american sign language. *Behavior Research Methods*, 14:467–474, 1982. 10.3758/BF03203314.
- [45] T. Starner, J. Weaver, e A. Pentland. Real-time american sign language recognition using desk and wearable computer based video. 20(12):1371–1375, 1998.
- [46] J. Sung, C. Ponce, B. Selman, e A. Saxena. Human activity detection from rgbd images. *CoRR*, abs/1107.0169, 2011. informal publication.
- [47] Ø. D. Trier, A. K. Jain, e T. Taxt. Feature extraction methods for character recognition - a survey, 1995.
- [48] Vladimir N. Vapnik. *Statistical learning theory*. Wiley, 1 edition, setembro de 1998.

- [49] C. Vogler e D. Metaxas. Asl recognition based on a coupling between hmms and 3d motion analysis. *Proc. Sixth Int Computer Vision Conf*, páginas 363–369, 1998.
- [50] C. Wang, W. Gao, e S. Shan. An approach based on phonemes to large vocabulary chinese sign language recognition. *Proc. Fifth IEEE Int Automatic Face and Gesture Recognition Conf*, páginas 411–416, 2002.
- [51] H. Wang, M. C. Leu, e C. Oz. American sign language recognition using multi-dimensional hidden markov models. *Journal of Information Science and Engineering*, 22(5), 2006.
- [52] Q. Wang, X. Chen, L. Zhang, C. Wang, e W. Gao. Viewpoint invariant sign language recognition. *Proc. IEEE Int. Conf. Image Processing ICIP 2005*, volume 3, 2005.
- [53] Z. Zafrulla, H. Brashear, T. Starner, H. Hamilton, e P. Presti. American sign language recognition with the kinect. *Proceedings of the 13th international conference on multimodal interfaces*, ICMI '11, páginas 279–286, New York, NY, USA, 2011. ACM.
- [54] Z. Zafrulla, H. Brashear, Pei Yin, P. Presti, T. Starner, e H. Hamilton. American sign language phrase verification in an educational game for deaf children. *Pattern Recognition (ICPR), 2010 20th International Conference on*, páginas 3846 –3849, aug. de 2010.

APÊNDICE A

EXEMPLOS DE IMAGENS 2D DAS VISÕES FRONTAL E LATERAL DA MÃO.

Este apêndice apresenta exemplos de imagens das visões frontal e lateral de cada configuração de mão da LIBRAS.

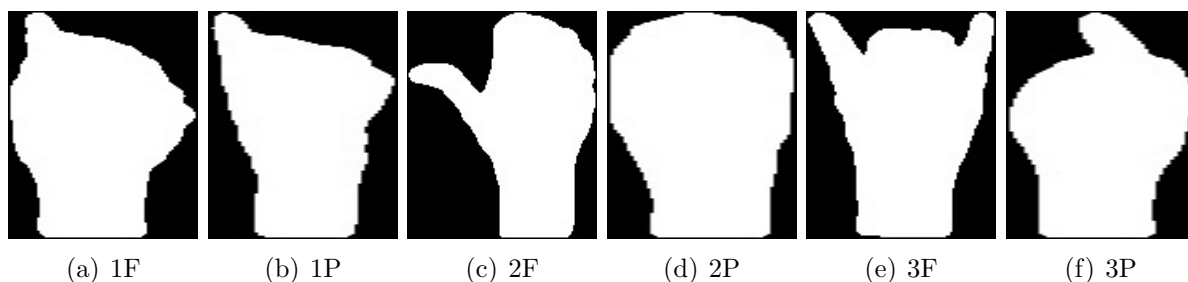


Figura A.1: Imagens das visões frontal (F) e perfil (P) das configurações de mão 1 (a,b), 2 (c,d) e 3 (e,f).

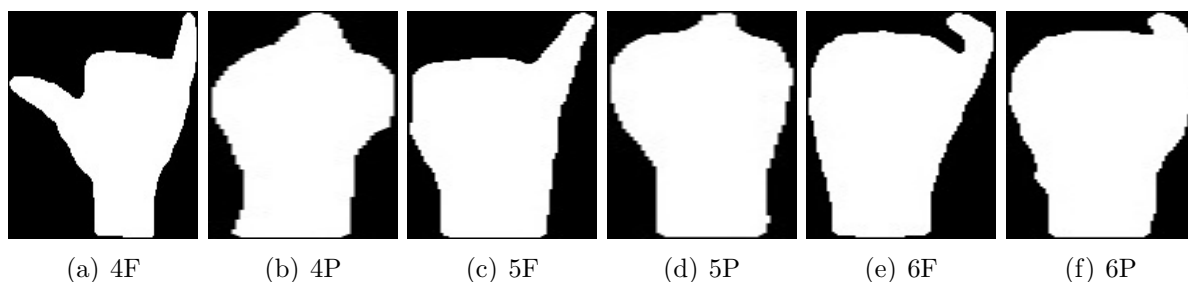


Figura A.2: Imagens das visões frontal (F) e perfil (P) das configurações de mão 4 (a,b), 5 (c,d) e 6 (e,f).

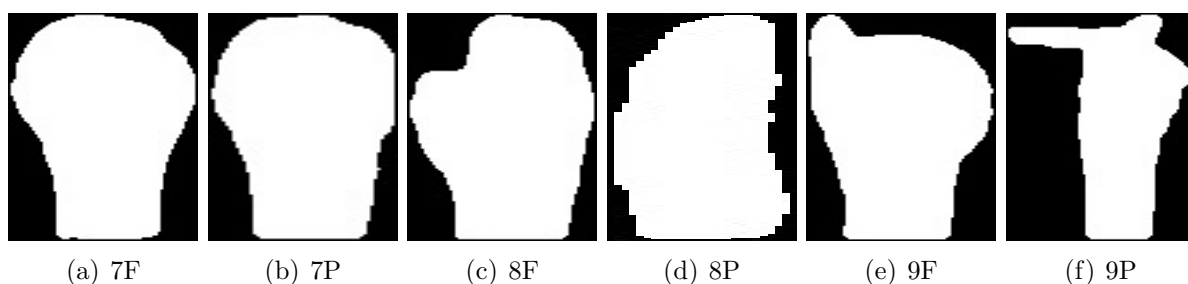


Figura A.3: Imagens das visões frontal (F) e perfil (P) das configurações de mão 7 (a,b), 8 (c,d) e 9 (e,f).

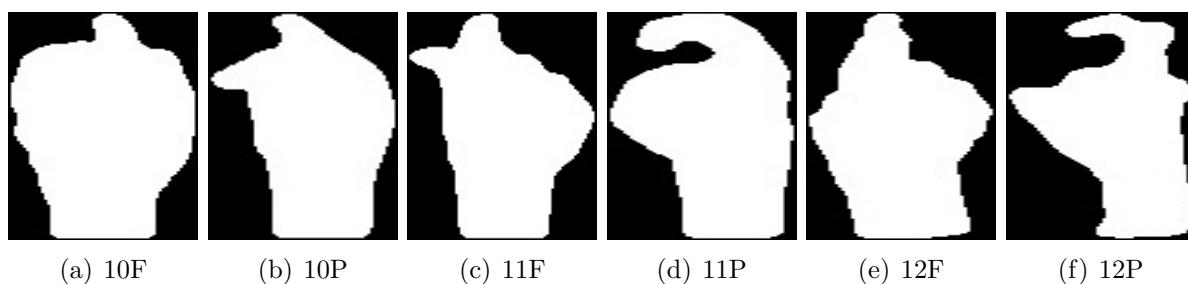


Figura A.4: Imagens das visões frontal (F) e perfil (P) das configurações de mão 10 (a,b), 11 (c,d) e 12 (e,f).

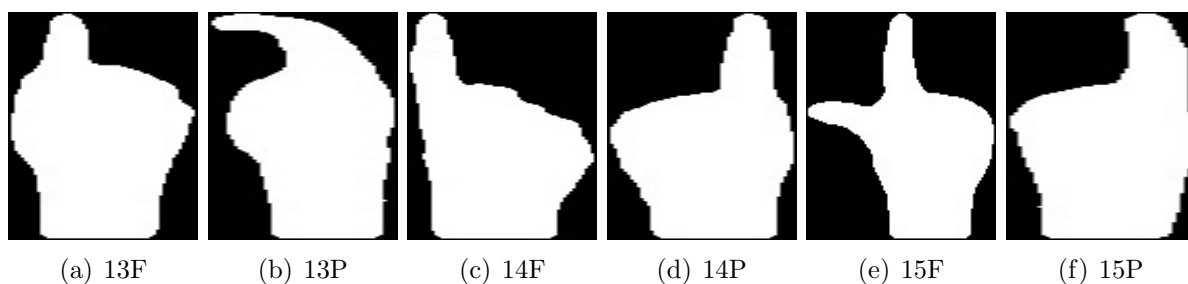


Figura A.5: Imagens das visões frontal (F) e perfil (P) das configurações de mão 13 (a,b), 14 (c,d) e 15 (e,f).

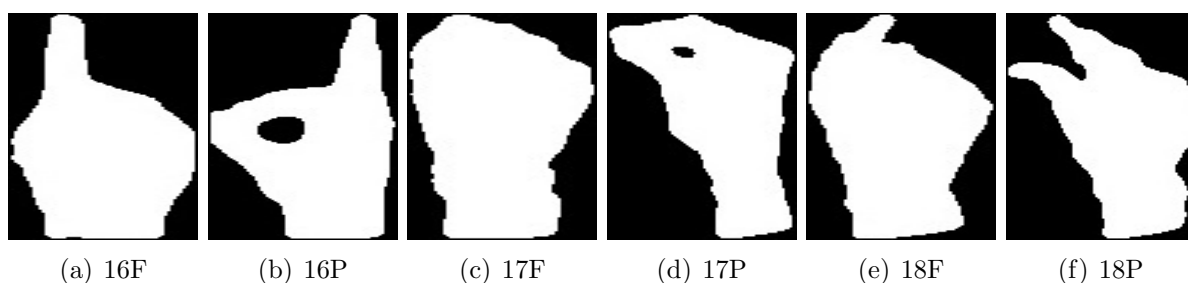


Figura A.6: Imagens das visões frontal (F) e perfil (P) das configurações de mão 16 (a,b), 17 (c,d) e 18 (e,f).

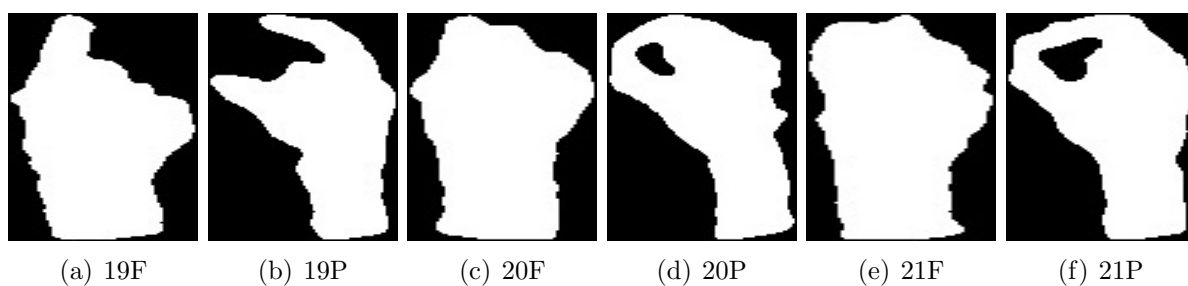


Figura A.7: Imagens das visões frontal (F) e perfil (P) das configurações de mão 19 (a,b), 20 (c,d) e 21 (e,f).

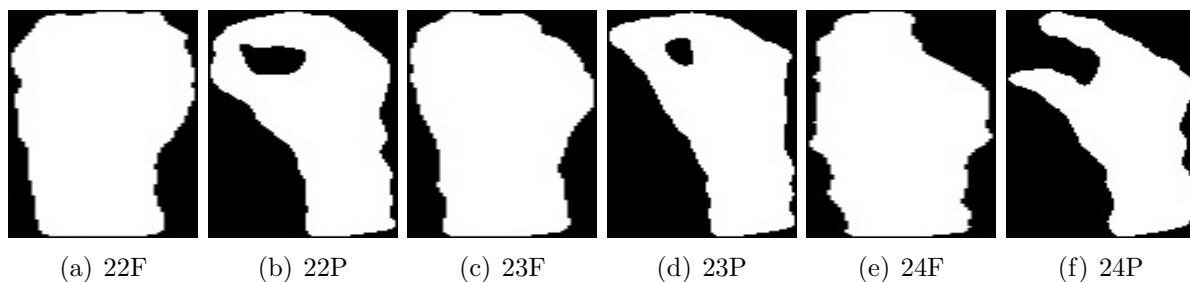


Figura A.8: Imagens das visões frontal (F) e perfil (P) das configurações de mão 22 (a,b), 23 (c,d) e 24 (e,f).

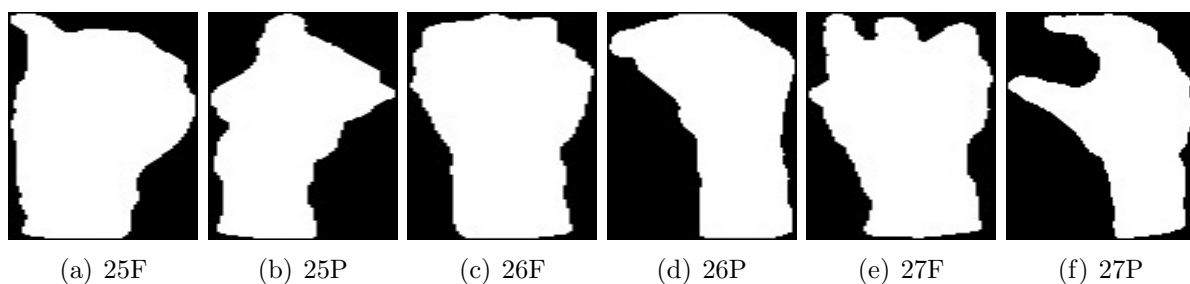


Figura A.9: Imagens das visões frontal (F) e perfil (P) das configurações de mão 25 (a,b), 26 (c,d) e 27 (e,f).

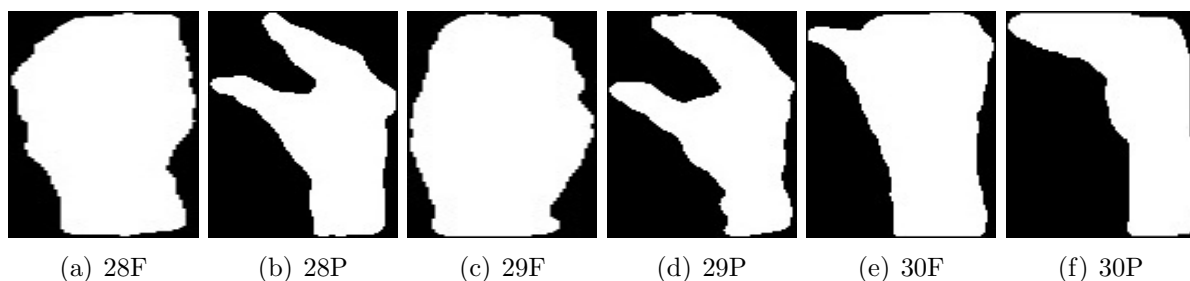


Figura A.10: Imagens das visões frontal (F) e perfil (P) das configurações de mão 28 (a,b), 29 (c,d) e 30 (e,f).

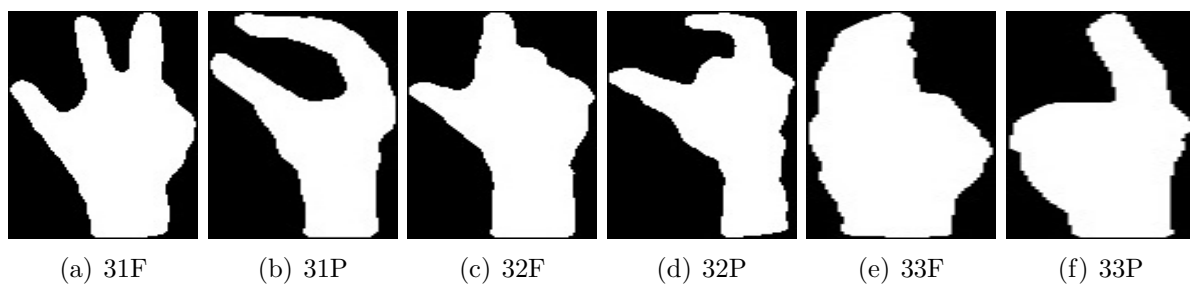


Figura A.11: Imagens das visões frontal (F) e perfil (P) das configurações de mão 31 (a,b), 32 (c,d) e 33 (e,f).

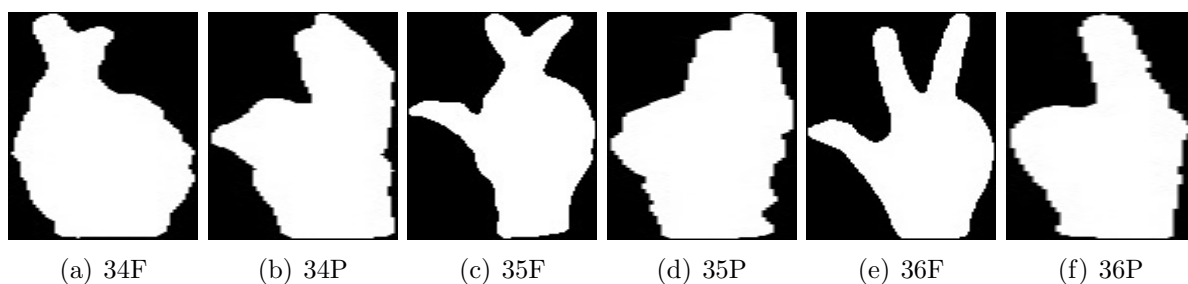


Figura A.12: Imagens das visões frontal (F) e perfil (P) das configurações de mão 34 (a,b), 35 (c,d) e 36 (e,f).

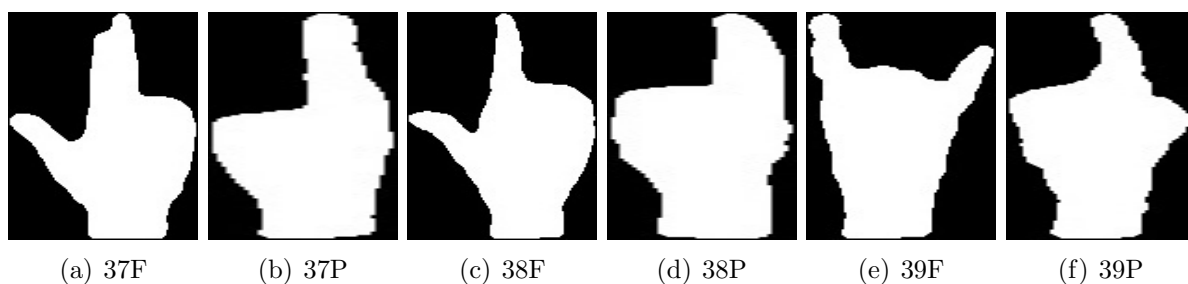


Figura A.13: Imagens das visões frontal (F) e perfil (P) das configurações de mão 37 (a,b), 38 (c,d) e 39 (e,f).

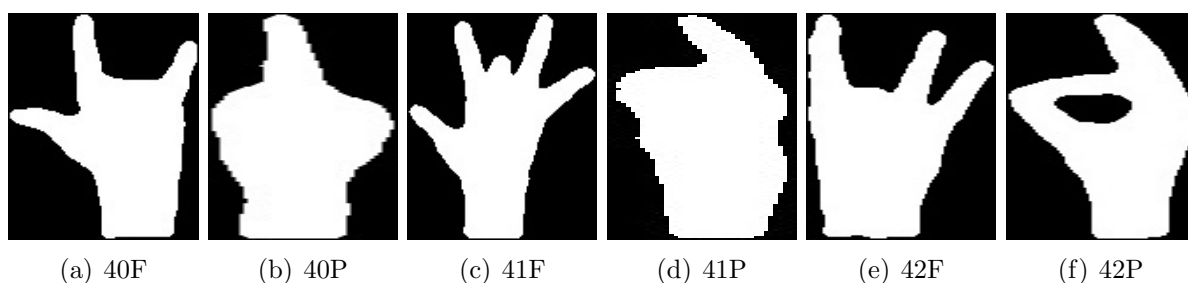


Figura A.14: Imagens das visões frontal (F) e perfil (P) das configurações de mão 40 (a,b), 41 (c,d) e 42 (e,f).

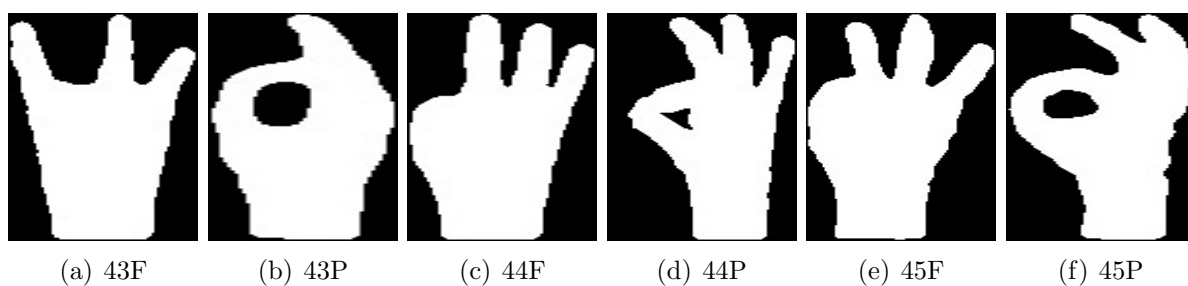


Figura A.15: Imagens das visões frontal (F) e perfil (P) das configurações de mão 43 (a,b), 44 (c,d) e 45 (e,f).

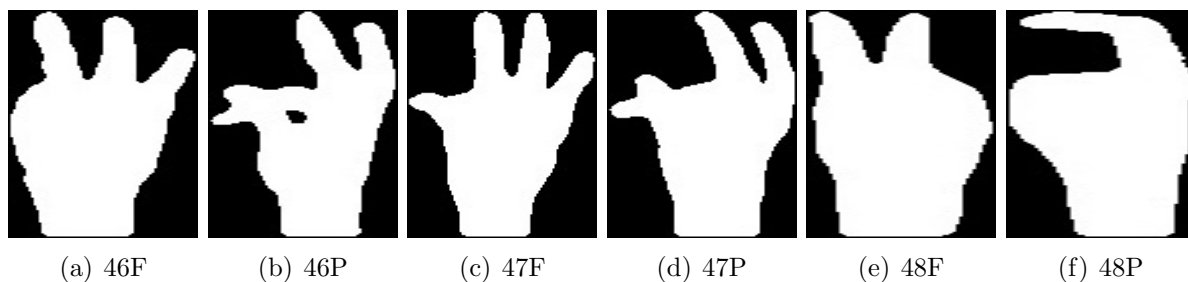


Figura A.16: Imagens das visões frontal (F) e perfil (P) das configurações de mão 46 (a,b), 47 (c,d) e 48 (e,f).

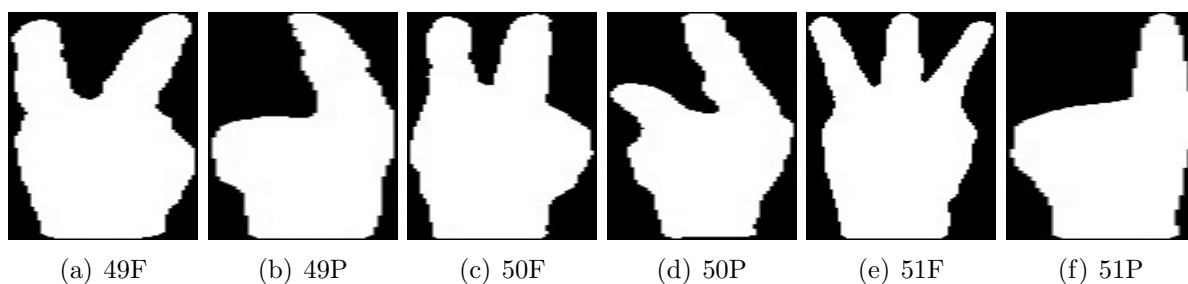


Figura A.17: Imagens das visões frontal (F) e perfil (P) das configurações de mão 49 (a,b), 50 (c,d) e 51 (e,f).

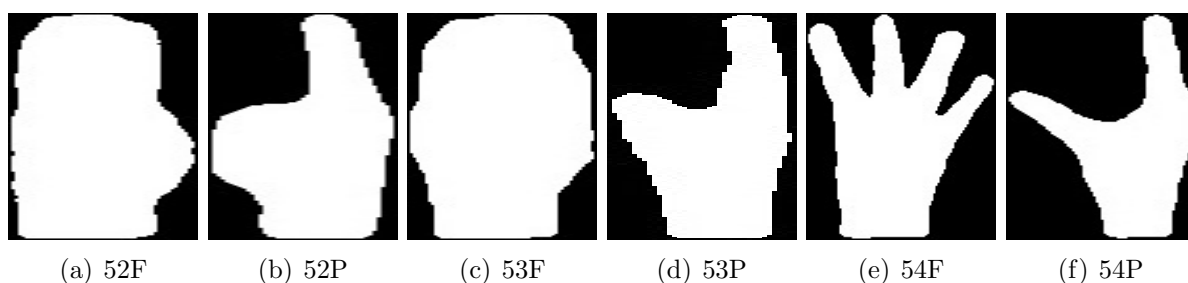


Figura A.18: Imagens das visões frontal (F) e perfil (P) das configurações de mão 52 (a,b), 53 (c,d) e 54 (e,f).

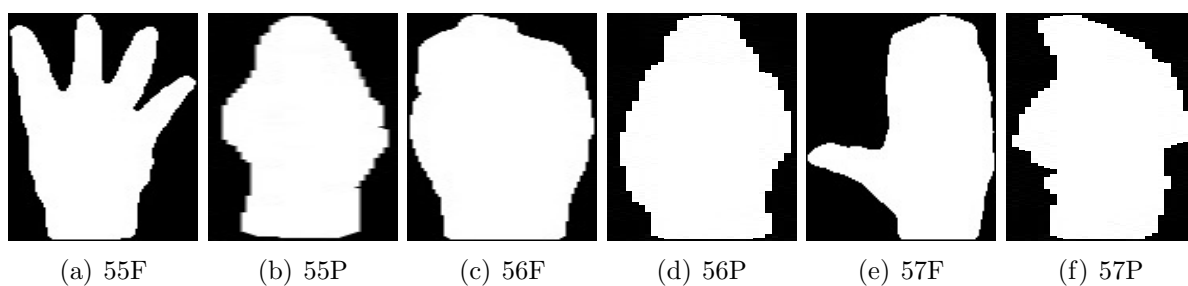


Figura A.19: Imagens das visões frontal (F) e perfil (P) das configurações de mão 55 (a,b), 56 (c,d) e 57 (e,f).

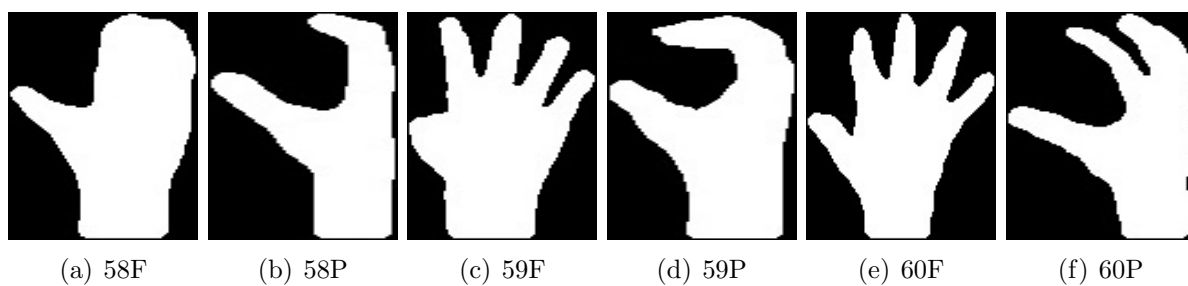


Figura A.20: Imagens das visões frontal (F) e perfil (P) das configurações de mão 58 (a,b), 59 (c,d) e 60 (e,f).

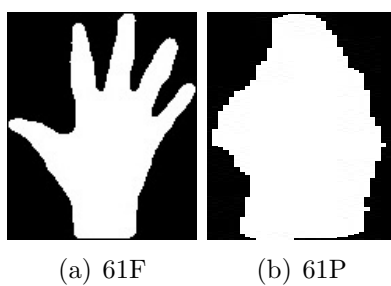


Figura A.21: Imagens das visões frontal (a/61F) e perfil (b/61P) da configuração de mão 61.

APÊNDICE B

EXEMPLOS DE MALHAS GERADAS POR SILHUETA

Este apêndice apresenta um exemplo de malha 3D gerada pelo método da silhueta para cada uma das 61 configurações de mão da LIBRAS.

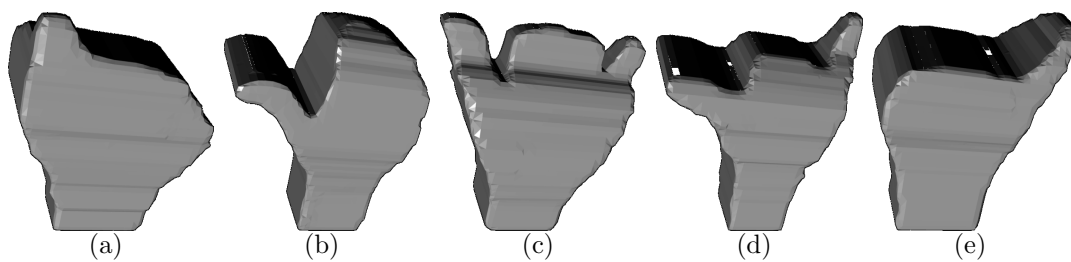


Figura B.1: Exemplos de malhas referentes às configuração de mão 1 (a), 2 (b), 3 (c), 4 (d) e 5 (e).

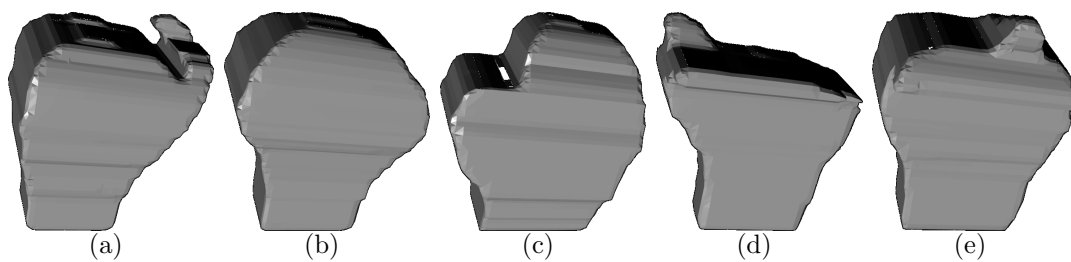


Figura B.2: Exemplos de malhas referentes às configuração de mão 6 (a), 7 (b), 8 (c), 9 (d) e 10 (e).

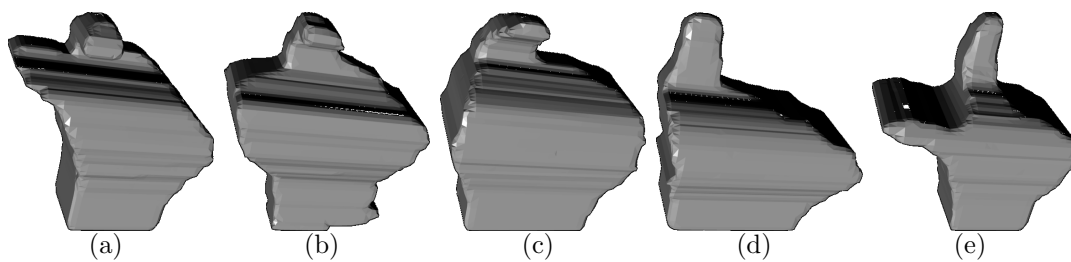


Figura B.3: Exemplos de malhas referentes às configuração de mão 11 (a), 12 (b), 13 (c), 14 (d) e 15 (e).

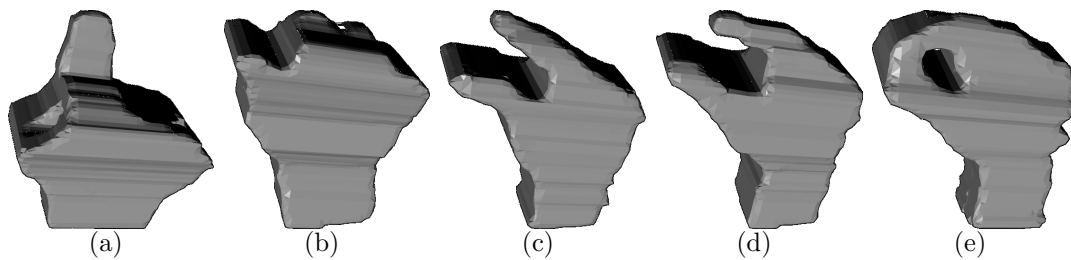


Figura B.4: Exemplos de malhas referentes às configuração de mão 16 (a),17 (b),18 (c),19 (d) e 20 (e).

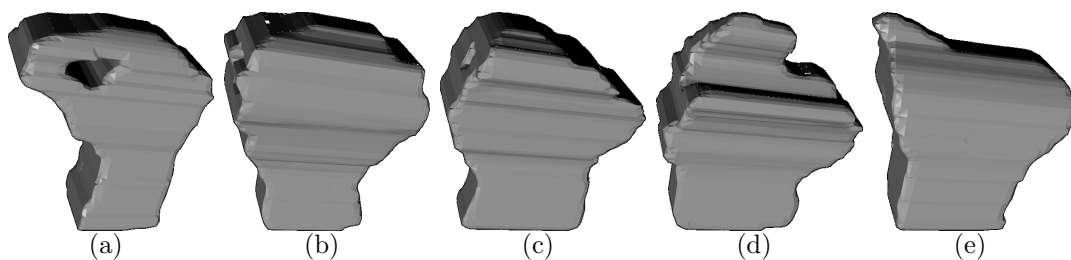


Figura B.5: Exemplos de malhas referentes às configuração de mão 21 (a),22 (b),23 (c),24 (d) e 25 (e).

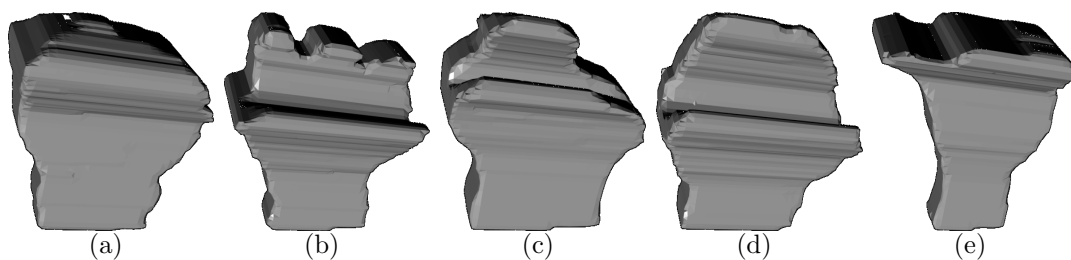


Figura B.6: Exemplos de malhas referentes às configuração de mão 26 (a),27 (b),28 (c),29 (d) e 30 (e).

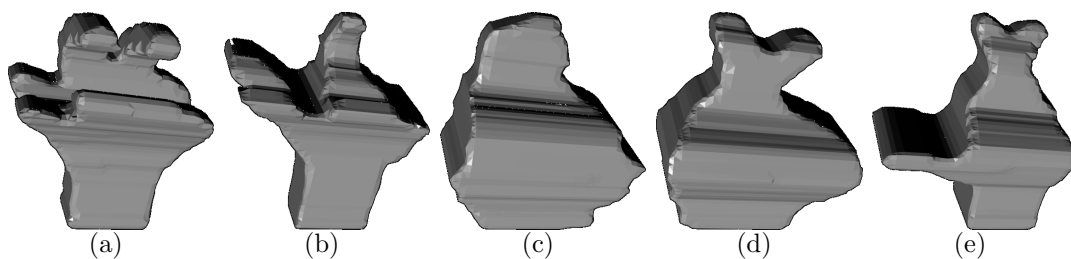


Figura B.7: Exemplos de malhas referentes às configuração de mão 31 (a),32 (b),33 (c),34 (d) e 35 (e).

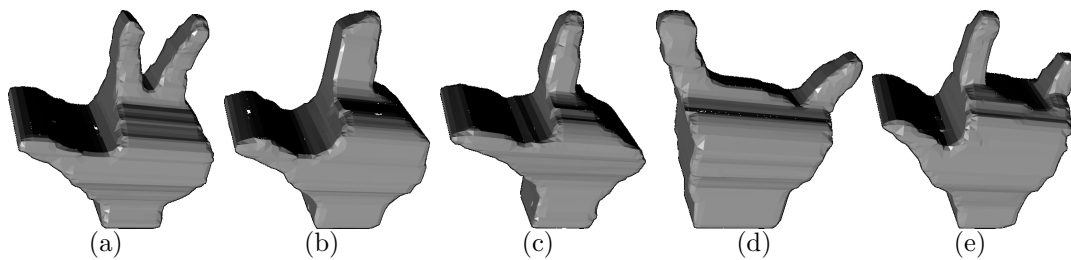


Figura B.8: Exemplos de malhas referentes às configuração de mão 36 (a),37 (b),38 (c),39 (d) e 40 (e).

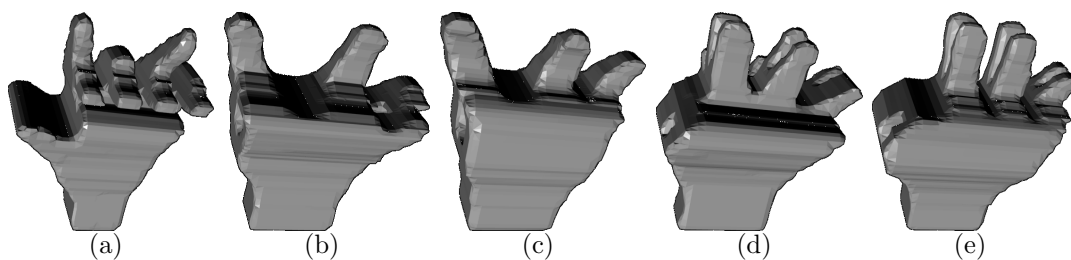


Figura B.9: Exemplos de malhas referentes às configuração de mão 41 (a),42 (b),43 (c),44 (d) e 45 (e).

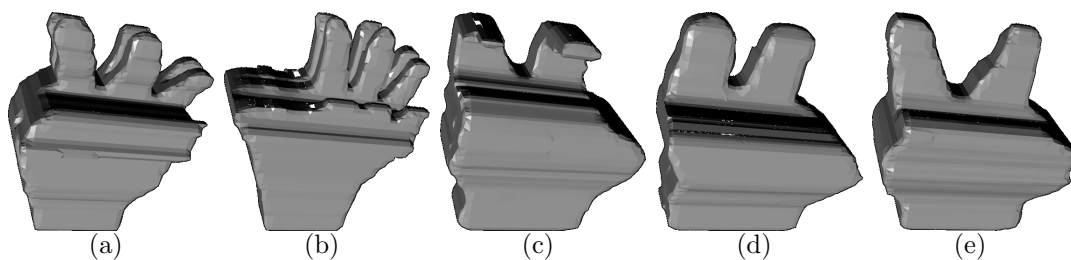


Figura B.10: Exemplos de malhas referentes às configuração de mão 46 (a),47 (b),48 (c),49 (d) e 50 (e).

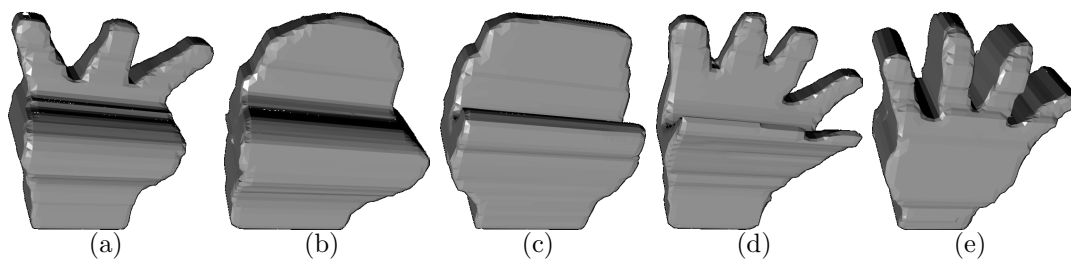


Figura B.11: Exemplos de malhas referentes às configuração de mão 51 (a),52 (b),53 (c),54 (d) e 55 (e).

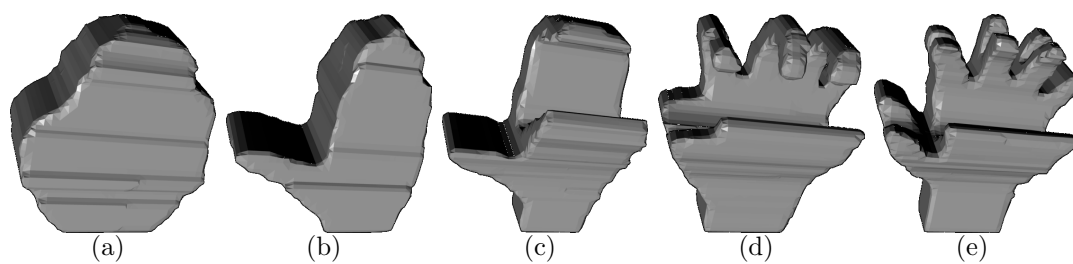


Figura B.12: Exemplos de malhas referentes às configuração de mão 56 (a), 57 (b), 58 (c), 59 (d) e 60 (e).

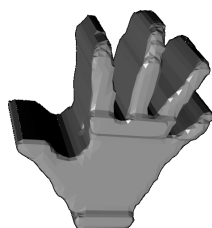


Figura B.13: Exemplo de malha referentes às configuração de mão 61.